Trabajo de Fin de Grado

Decodificación de información sobre el estado cognitivo de personas mayores a partir de mediciones de electroencefalograma durante el sueño

Information decoding about elderly people cognitive state from EEG activity during sleep

Autor/es

Óscar Brizuela García

Directora y co-director

María Sierra Torralba y Eduardo López-Larraz

Ponente

Javier Mínguez Zafra

Graduado en Ingeniería Informática

Especialidad en Sistemas de Información

Escuela de Ingeniería y Arquitectura

2024/2025

Resumen

El deterioro cognitivo es un problema de salud pública, causado por el envejecimiento de la sociedad, con un gran impacto en la calidad de vida y en los sistemas de salud. Estudios recientes destacan la relación entre los patrones de sueño y el desarrollo de trastornos cognitivos, como el Deterioro Cognitivo Leve o la demencia. En este contexto, la inteligencia artificial emerge como una herramienta prometedora, capaz de analizar grandes volúmenes de datos complejos, como las señales cerebrales (electroencefalogramas, EEGs), para tratar de identificar patrones y realizar predicciones que faciliten un diagnóstico precoz.

El objetivo de este proyecto es intentar decodificar el estado cognitivo de personas mayores en función de las señales EEG obtenidas durante el sueño, para poder clasificar estos sujetos en uno de los principales grupos que caracterizan el nivel de deterioro cognitivo: personas sanas, con Quejas Subjetivas de Memoria (SCI), con Deterioro Cognitivo Leve (MCI) o con demencia (DEM). Esta tarea de clasificación ha sido llevada a cabo mediante el entrenamiento de múltiples modelos y arquitecturas tanto de *Machine Learning* (ML) como de *Deep Learning* (DL), con el objetivo de conseguir las mejores métricas posibles sobre un dataset adquirido por el equipo de Neurociencia de *Bitbrain*. Las señales EEG analizadas han sido obtenidas a través del dispositivo *Ikon Sleep*, desarrollado por la misma empresa, el cual es colocado de forma autónoma (o con la supervisión de algún familiar o cuidador) por los sujetos en sus hogares.

Este problema de clasificación se ha abordado de 3 formas distintas. En primer lugar, se han tratado las sesiones como la evolución de las fases del sueño a lo largo del tiempo (hipnogramas), obtenida a partir de las señales EEG mediante un modelo externo de DL que clasifica cada ventana de 30 segundos de EEG en una fase del sueño determinada (W, N1, N2, N3 ó REM). La clasificación de estas secuencias se ha realizado utilizando Redes Neuronales Recurrentes (RNNs). En segundo lugar, se ha eliminado la dimensión temporal de estos hipnogramas, obteniendo características (de ahora en adelante, features) derivadas directamente de la macroestructura del sueño. La clasificación de las sesiones caracterizadas como estas listas de features se ha llevado a cabo mediante diversos modelos de ML y DL, tanto lineales como no lineales. En tercer lugar se han caracterizado las sesiones como listas de features obtenidas a raíz de la microestructura (señales EEG). Estas features, agrupadas por las distintas fases del sueño, proporcionan una información más detallada acerca de la forma en la que duerme una persona. Además, se combinan con aquellas features derivadas de la macroestructura. Tras ello, se han llevado a cabo varios procesos de selección de features mediante estudios estadísticos y coeficientes de importancia. En esta forma de abordar el problema, la cual ha reportado los mejores resultados, han intervenido los mismos modelos de ML y DL utilizados con las features derivadas de la macroestructura.

Todos los experimentos se realizaron para tres configuraciones de etiquetado: la configuración 1 divide los datos en sesiones de pacientes sanos frente a las demás; la configuración 2 agrupa a sanos y SCI frente a MCI y DEM; y la configuración 3 separa las cuatro clases. Así, las configuraciones 1 y 2 son problemas de clasificación binaria, mientras que la configuración 3 es multiclase.

Como resultado, la combinación de features de la micro y macroestructura ha conseguido un 87.68 % de *accuracy* para la configuración 2, un 73.53 % para la configuración 1 y un 47.47 % para la 3.

En conclusión, en este trabajo se ha comprobado que la actividad EEG durante el sueño permite obtener información relevante acerca del estado cognitivo de personas mayores, lo que podría utilizarse en el futuro para mejorar sistemas de diagnóstico.

Índice general

Introducción

La demencia es un término general que se utiliza para describir un conjunto de síntomas relacionados con un deterioro cognitivo progresivo que afecta la memoria, el pensamiento, el comportamiento y la capacidad para realizar actividades cotidianas. No es una enfermedad específica, sino un síndrome que puede ser causado por diversas condiciones, siendo la enfermedad de Alzheimer (AD) la forma más común [1]. Las personas con demencia pueden experimentar dificultades para recordar información reciente, mantener conversaciones, resolver problemas o realizar tareas cotidianas. Además, pueden mostrar cambios en su personalidad, comportamiento y estado de ánimo, lo que puede afectar sus relaciones con familiares y amigos. La demencia no solo afecta a la persona diagnosticada, sino que también tiene un impacto significativo en sus cuidadores y seres queridos, quienes pueden enfrentarse a desafíos emocionales, físicos y financieros. El cuidado de personas con demencia requiere un enfoque multidisciplinario que incluya atención médica, apoyo psicológico y programas de intervención para mejorar la calidad de vida del paciente y su entorno.

De acuerdo a diversos estudios [2, 3], la forma en la que las personas duermen podría servir como indicador para determinar el estado cognitivo de las personas. Así como las personas sanas tienden a dormir de una determinada manera, las personas que sufren alguna enfermedad degenerativa también cuentan con determinadas características en su sueño. Por ello, para este trabajo se han utilizado electroencefalogramas (de ahora en adelante, EEGs) de personas mayores medidos durante el sueño. Estos EEGs se han obtenido mediante el dispositivo *Ikon Sleep*, una banda textil que se coloca fácilmente en la cabeza para medir la actividad cerebral del área frontal del cerebro mientras la persona duerme. De esta forma, estos sujetos se colocan el dispositivo de manera autónoma (o con ayuda de un familiar o cuidador) y sin supervisión de especialistas, empiezan a grabar los datos al principio de la noche (que, de ahora en adelante, llamaremos sesión) y detienen la grabación cuando se despiertan por la mañana.

Habitualmente, este tipo de datos se obtienen en entornos clínicos mediante estudios de polisomnografía (PSGs), donde no sólo se registra el EEG, sino también otras señales biológicas como el movimiento ocular, la actividad muscular y otros parámetros fisiológicos. Sin embargo, este proceso es caro (debido a los dispositivos utilizados [4]) y requiere de expertos, tanto para poder colocar el polisomnógrafo adecuadamente como para, tras las mediciones, determinar en qué fase del sueño se encuentra en cada momento el paciente de acuerdo a los datos recopilados.

A lo largo de este proyecto, hablaremos de dos tipos de estructuras del sueño fundamentales: microestructura y macroestructura.

La macroestructura, o arquitectura del sueño, es la organización general de las fases y ciclos del sueño durante una noche de descanso. Esta arquitectura es un patrón regular y cíclico que se repite varias veces a lo largo de la noche, con variaciones en la duración de las diferentes fases. No obstante, en escenarios reales y, sobre todo, en sesiones de personas mayores, la macroestructura no es tan regular, cíclica y, en términos generales, ideal como se desearía en la teoría. Cada periodo de 30 segundos es caracterizado como perteneciente a una fase del sueño determinada, determinando si el paciente se encuentra despierto o en vigilia (W), en una de las 3 fases No REM (N1, N2 ó N3) o en fase REM.

La microestructura trata patrones más detallados dentro de las fases del sueño, en particular los cambios que ocurren a nivel de actividad cerebral y fisiológica a lo largo de cada fase, observables en el EEG. Los EEGs utilizados en este proyecto han sido grabados a una frecuencia de sampleo de 128 Hz, de forma que cada ventana de 30 segundos cuenta con 3840 valores, midiendo así la

amplitud de la señal en microvoltios (μV).

También hablaremos de configuraciones de etiquetado, puesto que no solo se ha abordado el problema como una clasificación multiclase entre los cuatro grupos, sino también como una clasificación binaria. Las 3 configuraciones tratadas en este trabajo tratan a los pacientes sanos frente a los demás (configuración 1), a los sanos y con SCI frente a aquellos con MCI o DEM (configuración 2) y a los cuatro grupos por separado.

Dada la novedad en la forma de afrontar este proyecto (solo se ha encontrado un estudio que tratase de extraer información cognitiva de EEG durante el sueño), se ha optado por explorar diferentes vías para analizar las señales (a las que denominaremos "Approaches" a lo largo de todo el proyecto), y así comprobar cuál proporciona más información. Posteriormente se han desarrollado diversos algoritmos de ML y DL con el objetivo de clasificar estas sesiones en una de las clases descritas. Por tanto, este trabajo no solo tiene como intención desarrollar modelos capaces de clasificar las sesiones, sino que también busca utilizar la información disponible para extraer qué características del sueño son las más importantes a la hora de determinar el estado cognitivo del paciente.

Además, hay que tener en cuenta que la demencia, y las enfermedades que la causan, son el resultado de procesos degenerativos complejos de años de duración. Estas patologías actualmente no tienen cura, y los tratamientos farmacológicos que se aplican no son excesivamente efectivos. La razón más probable es que suelen empezar a aplicarse demasiado tarde, cuando la persona ya tiene un deterioro cognitivo algo avanzado. La corriente actual es tratar de hacer una detección temprana de estos procesos, para tratar de empezar a actuar antes, esperando así que los tratamientos sean más efectivos. En este trabajo se detallará el desarrollo de las metodologías implementadas para intentar ayudar en una detección temprana del deterioro cognitivo.

Objetivos y alcance, herramientas y cronograma

Objetivos y alcance

Los objetivos propuestos que se intentarán alcanzar durante la realización de este trabajo son los siguientes:

1. Diseñar un modelo que, dada la grabación de EEG de un paciente durante la noche, consiga métricas interesantes para servir de base a estudios futuros acerca de la relación entre el estado cognitivo y la demencia.
2. Explorar la relación entre distintas características del sueño y el estado cognitivo de un paciente.

Además, este trabajo contribuye a la consecución de los Objetivos de Desarrollo Sostenible (ODSs) 3.4 y 3.d, relacionados con la salud y el bienestar, 8.2, relacionado con el trabajo decente y el crecimiento económico, y 9.c, relacionado con la industria, la innovación y las infraestructuras.

Herramientas y materiales

Para la realización de experimentos que han requerido largos tiempos de entrenamiento al utilizar modelos de DL debido a la complejidad del modelo o al alto número de datos de entrenamiento, se ha utilizado la GPU del departamento de Data Science de la empresa. Esta unidad de procesamiento gráfico, de marca *Nvidia*, corresponde al modelo *GeForce RTX 3080 Ti*. Las especificaciones del motor GPU utilizado para estas tareas son las presentes en la Tabla 1:

Núcleos NVIDIA CUDA	10240
Frecuencia de reloj acelerada (GHz)	1.67
Frecuencia de reloj normal (GHz)	1.37
Configuración de memoria estándar	12 GB GDDR6X
Ancho de la interfaz de memoria	384 bits

Tabla 1: Especificaciones de la GPU de la empresa utilizada en los experimentos

Por otro lado, para aquellas tareas de preprocesamiento de datos con el fin de obtener features a partir de los EEGs que requerían grandes tiempos de cómputo, se ha utilizado la CPU del mismo equipo. Esta unidad de procesamiento cuenta con las características de la Tabla 2:

Número de potenciales clientes finales diarios	5
Memoria RAM	64 GB
Procesador y capacidad de proceso	Intel Core i9-10900K CPU @ 3.70GHz
Espacio en disco	5 TB
Ancho de banda	1 TB
Licencia del Sistema Operativo	Ubuntu 22.04

Tabla 2: Especificaciones de la CPU de la empresa utilizada en los experimentos

Las pruebas que no han requerido de una gran cantidad de recursos de cómputo han sido realizadas sobre el equipo personal del autor del trabajo, un ordenador portátil *Lenovo IdeaPad 3 15IIL05*, cuyas especificaciones se describen en la Tabla 3:

Procesador	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Memoria RAM	8,00 GB
Espacio en disco	512 GB
Licencia del Sistema Operativo	Windows 11 Home

Tabla 3: Especificaciones del equipo del autor utilizado en los experimentos

Los datos correspondientes a los EEGs han sido medidos con el dispositivo *Ikon Sleep* (Figura 1), desarrollado por la empresa *Bitbrain*. Este dispositivo, integrado en una banda textil, cuenta con dos canales para medir los EEGs en las zonas anterior frontal izquierda y derecha (denominados *AF7* y *AF8*, respectivamente, como se muestra en la Figura 2), que se encuentran en contacto directo con el cuero cabelludo. Los datos utilizados en este proyecto provienen de personas mayores, quienes se han colocado autónomamente (o con algún familiar o cuidador) el dispositivo en sus hogares, sin supervisión de especialistas. Ambas imágenes son cortesía de *Bitbrain*.

Figura 1: Ejemplo de utilización del dispositivo *Ikon Sleep*

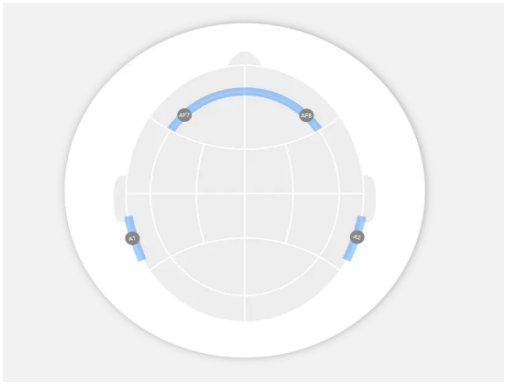


Figura 2: Posicionamiento de los canales del dispositivo *Ikon Sleep*

A lo largo de todo el proyecto, el lenguaje de programación utilizado ha sido *Python*, en su versión 3.12.3. El tratamiento de los distintos datasets se ha llevado a cabo con la librería *pandas*, que permite un uso sencillo de estos conjuntos en forma de DataFrames (tablas bidimensionales), y la librería *NumPy*, que facilita la manipulación de estructuras de datos multidimensionales, muy presentes en todo el proyecto. Por otro lado, en cuanto a los modelos utilizados para las distintas clasificaciones, así como para la división de los datasets en diferentes *splits*, se ha hecho uso de la librería *Scikit-Learn* para los modelos de ML y de la API *Keras*, de *Tensorflow*, para los modelos de DL.

Para obtener varias de las features descritas a lo largo de este trabajo, así como para mostrar el hipnograma de la Figura 5, se ha hecho uso de la librería *YASA* (*Yet Another Spindle Algorithm*).

La herramienta en la que se ha ejecutado mayoritariamente este código ha sido *Jupyter Notebook*, cuyos ficheros cuentan con la extensión *.ipynb*. Esta herramienta de código abierto permite

ejecutar el código de forma interactiva y por partes, en distintas celdas, lo que es crucial para llevar a cabo de manera sencilla y correcta todas las fases de la pipeline implementada (carga de datos, preprocesamiento, *feature engineering*, entrenamiento, visualización de resultados...). No obstante, cuando ha sido necesaria la utilización del equipo de la empresa, cuyo código se ejecuta de manera sencilla a través de la línea de comandos en un entorno *Linux*, estos ficheros *.ipynb* se transformaron a scripts *.py*, de *Python*.

Cronograma

En el diagrama de Gantt de la Figura 3 se muestra la planificación temporal del proyecto a lo largo de las 18 semanas, detallando las tareas principales y sus duraciones. Se ha dividido en función de las 3 formas de abordar el problema del proyecto. La granularidad del cronograma es en semanas.

Comienzo del proyecto:	9/17/2024 (martes)
Finalización del proyecto:	1/26/2025 (domingo)

TAREA	COMIENZO	FIN	DÍAS	Nº de SEMANAS
DOCUMENTACIÓN				
Introducción a los EEGs e hipnogramas	mar 9/17/24	lun 9/23/24	7	1
Investigación sobre proyectos previos sobre sueño y deterioro cognitivo (Estado del Arte)	sáb 9/21/24	sáb 1/18/25	120	17
Documentación sobre modelos de ML y DL a utilizar	mié 9/25/24	vie 11/08/24	45	6
Memoria final	sáb 10/05/24	dom 1/26/25	114	16
Estudio de la macroestructura como secuencia de fases del sueño				
Procesamiento de datos sintéticos y reales	mié 9/25/24	sáb 9/28/24	4	1
Realización de experimentos y anotaciones de resultados	vie 10/11/24	mié 10/23/24	13	2
Estudio de la macroestructura como features derivadas de los hipnogramas				
Cálculo y exploración de features de la macroestructura a raíz de la secuencia de fases del sueño	lun 10/28/24	dom 11/03/24	7	1
Procesamiento de datos sintéticos y reales	lun 11/04/24	mar 11/05/24	2	1
Realización de experimentos y anotaciones de resultados	lun 11/18/24	sáb 11/30/24	13	2
Combinación de microestructura y macroestructura				
Cálculo y exploración de features de la microestructura a raíz de las señales EEG	lun 12/02/24	lun 12/16/24	15	2
Procesamiento de datos sintéticos y reales	jue 12/05/24	mié 12/18/24	14	2
Realización de experimentos y anotaciones de resultados	jue 12/26/24	dom 1/12/25	18	3
Definición y diseño de modelos ML y DL				
Definición y diseño de arquitecturas RNNs	vie 9/27/24	mar 10/01/24	5	1
Definición y diseño de modelos ML y DL	mié 11/06/24	vie 11/08/24	3	1

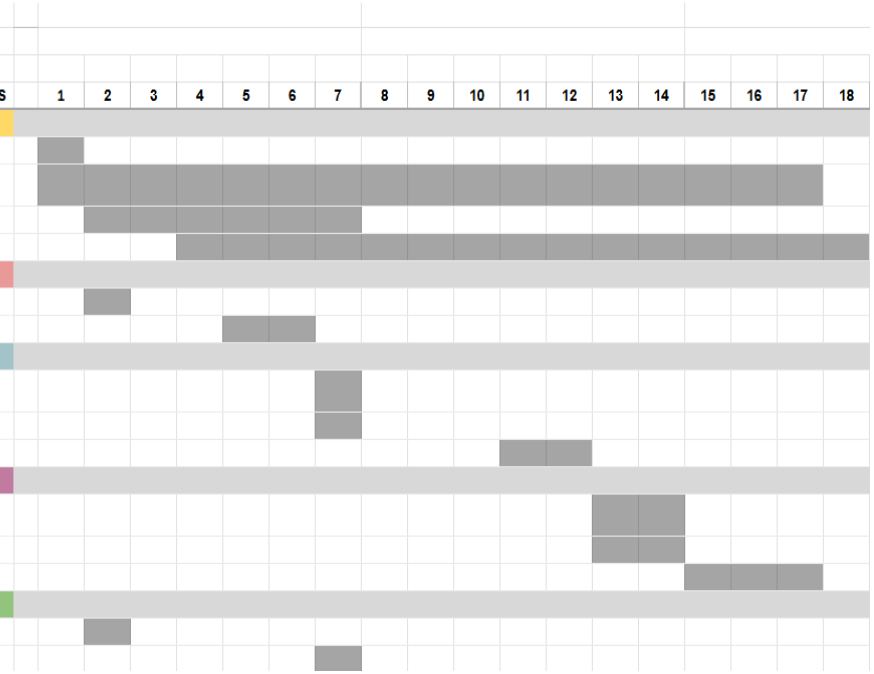


Figura 3: Cronograma de Gannt

En total, el autor de este proyecto ha invertido 468.5 horas, que se han ido anotando diariamente. De ellas, 367.5 horas corresponden a la construcción del código del proyecto, y 101 horas a la realización de esta memoria.

Estado del arte

Debido al creciente número de patologías que asocian el envejecimiento con el deterioro cognitivo, el estudio de los EEGs ha ganado relevancia en los últimos años. Sin embargo, a pesar de que las investigaciones en el campo de las señales cerebrales ha aumentado considerablemente, el número de estudios que tratan los EEGs medidos durante el sueño es una pequeña proporción. Muchos de estos se encuentran presentes en la revista SLEEP, una de las revistas de investigación internacionales más prestigiosas en el campo del sueño y las enfermedades relacionadas con este, entre otros aspectos.

El trabajo más similar y completo hasta la fecha [3] realiza un estudio con 10784 mediciones de PSG de 8044 pacientes diferentes, de los cuales el 89.49 % de las sesiones corresponden a pacientes sanos, el 6.33 % corresponde a pacientes con MCI y el 4.21 % a pacientes con demencia. Este trabajo utiliza diversos algoritmos de ML para clasificaciones binarias: personas sanas frente a personas con demencia, obteniendo un AUROC de 0.78 con el modelo SVM; personas sanas frente a MCI, consiguiendo un AUROC de 0.73; y personas con MCI frente a personas con demencia, logrando un AUROC de 0.76. Gracias a este estudio, podemos concluir que features relacionadas con ondas lentas y spindles en N2, y features relacionadas con las bandas Θ y δ en las fases W y N1 serían útiles para diferenciar MCI y DEM. De manera similar, features relacionadas con la actividad α en W y N1 y diversas features de la fase REM permiten diferenciar aquellos pacientes sanos de aquellos con un deterioro cognitivo.

También existen estudios que investigan acerca de las alteraciones del EEG de pacientes con MCI y AD, quienes tienen altas latencias (les cuesta más dormirse desde que se echan a la cama), además de contar con menos ondas lentas y más cambios en la actividad δ respecto a los sujetos sanos [5].

Existe otro estudio [6], con objetivo de diferenciar entre personas con MCI y sanas (20 sesiones de cada clase, obtenidas con PSGs) en función de su microestructura del sueño. Utilizan, además de otras features, métodos de obtención de dimensión fractal y entropía, consiguiendo una *accuracy* del 93.46 % mediante una Red Neuronal Recurrente (en concreto, una red GRU).

Otra investigación [7], cuyo objetivo es detectar el insomnio utilizando un dataset multimodal, reporta una *accuracy* del 95.83 % para diversos modelos con árboles de decisión, y del 91.67 % para modelos como SVM o Regresión Logística.

Por otro lado, la gran mayoría de estudios relacionados con el sueño simplemente describen características concretas de la microestructura o macroestructura, sin tratar de clasificar propiamente los sujetos. El artículo [8] lista 74 estudios que utilizan diversos modelos de DL entre 2013 y 2024 para analizar EEGs. Un ejemplo de ello es el artículo [9], que compara las diferencias en los EEG de personas con MCI y AD estando dormidas y despiertas, pero sin llegar a tratar de clasificarlas automáticamente.

A pesar de que muchos de estos estudios no se centran en la explicabilidad de los modelos utilizados, sí que realizan diversas técnicas de extracción y selección de features. Las features reportadas como "importantes" en estos estudios han servido de pequeña guía para acotar el problema abordado en este trabajo.

Metodología

Dataset

Para este proyecto se ha trabajado con un novedoso dataset adquirido por el equipo de Neurociencia de *Bitbrain*, en el marco de un proyecto de investigación que busca desarrollar herramientas que ayuden a mejorar el diagnóstico y tratamiento temprano de enfermedades relacionadas con el envejecimiento. El objetivo específico que se busca con este dataset es validar un instrumento de uso domiciliario (la banda textil *Ikon Sleep*) para la cuantificación de la función cognitiva en una población en riesgo de demencia. A pesar de que a lo largo de todo el proyecto el número de pacientes (y, por tanto, de sesiones) fue aumentando gradualmente con el tiempo, todos los experimentos finales se realizaron con un total de 219 sesiones. Estas sesiones pertenecen a sujetos con 4 estados cognitivos diferentes. Cabe mencionar que el deterioro cognitivo es un proceso continuo que no presenta unos límites claros entre clases. Cada categoría implica una compleja interacción de factores objetivos y subjetivos. De esta manera, las evaluaciones objetivas (como los tests cognitivos) buscan medir el rendimiento cognitivo con precisión, pero pueden no captar pequeños matices entre diferentes estados cognitivos.

En primer lugar, el dataset cuenta con sesiones de sujetos sanos (*Healthy* - H). Estas personas no tienen ningún tipo deterioro cognitivo detectado más allá del causado normativamente por la propia edad. Servirán como grupo de control.

En segundo lugar, tenemos sesiones de pacientes con Quejas Subjetivas de Memoria (*Subjective Cognitive Impairment* - SCI). Estos pacientes perciben (por tanto, subjetivamente) problemas para recordar o retener información, aunque estos problemas no afecten en su vida cotidiana. Por lo general, son personas que han tenido educación a lo largo de sus vidas y que, por tanto, no entran en las clases de deterioro cognitivo debido a que sus puntuaciones en los tests utilizados para realizar estos diagnósticos son demasiado altas para caracterizarlos con un deterioro.

En tercer lugar, se cuenta con personas con Deterioro Cognitivo Leve (*Mild Cognitive Impairment* - MCI). Los pacientes diagnosticados con MCI se encuentran en una etapa intermedia entre el deterioro previsto de la memoria y el pensamiento, que sucede con la edad, y el deterioro más grave de la demencia. Puede incluir problemas de memoria, de lenguaje o de capacidad de juicio. Su condición no les interfiere en el día a día, ya que tienen independencia a base de estrategias. Al ser la etapa anterior a la demencia total, es importante diagnosticar correctamente a estas personas, para poder tratarlas cuanto antes (con personas de asistencia o medicación) antes de que su estado cognitivo empeore.

En último lugar, encontramos los pacientes con demencia. Esta afección neurodegenerativa afecta principalmente a las funciones cognitivas, como la memoria, el lenguaje, el razonamiento y la capacidad para realizar actividades cotidianas. Los síntomas más comunes incluyen pérdida progresiva de la memoria, confusión, dificultades para comunicarse, cambios en la personalidad y alteraciones en el comportamiento.

Es importante remarcar que los pacientes caracterizados como pertenecientes a una clase determinada no han seguido un diagnóstico médico. Las etiquetas de estado cognitivo correspondientes a cada paciente han sido adjudicadas tras la realización de una serie de tests, observaciones y entrevistas a los pacientes en las instalaciones de la empresa, llevados a cabo por profesionales y siguiendo los criterios oficiales *DSM-5* para MCI y DEM [10] de diagnóstico en esta área. Para las personas con SCI no existe un test específico que los pueda diagnosticar [11], por lo que los

experimentadores de la empresa realizan una serie de preguntas binarias para clasificarlos como tales. Por tanto, la empresa no realiza disgnósticos médicos, solo estimaciones de acuerdo a dichos criterios.

La distribución de las 4 clases (sobre el número total de sesiones) que representan los estados cognitivos de los sujetos se puede visualizar en la Figura 4, la cual muestra un gran desbalanceo entre dichas clases:

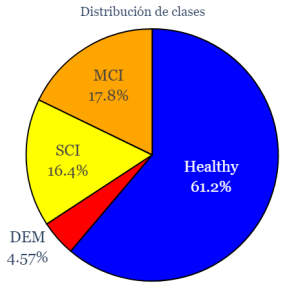


Figura 4: Distribución (%) de los estados cognitivos de las sesiones de los sujetos.

También fue facilitada información acerca de las puntuaciones en diversos tests cognitivos realizados por los pacientes, así como su edad. A pesar de que se visualizaron las diferencias entre estas variables para cada uno de los estados cognitivos, esta información finalmente no se utilizó (salvo en el modelo híbrido, el cual se detallará más adelante, cuyos resultados no fueron mejores que sin usar esta información). Sin embargo, sí se realizó un pequeño estudio de cómo la información de estos tests mejoraba considerablemente el rendimiento de los modelos de ML. Un pequeño ejemplo de ello se encuentra en el Anexo D, utilizando el modelo Random Forest Classifier.

La asociación entre el declive cognitivo y el sueño hacen que sea un objetivo prometedor para el diagnóstico y la terapia tempranos. Por tanto, la macro y microestructura son importantes biomarcadores para estos trastornos. Es por ello que ambas estructuras son calculadas utilizando los datos del registro de EEG y presentadas como entrada a los modelos, como veremos más adelante.

Estudio de la macroestructura

La macroestructura del sueño se refiere a la organización general de los ciclos y etapas del sueño a lo largo de la noche. Esta estructura se suele representar en hipnogramas, es decir, en secuencias de fases del sueño. Los hipnogramas de cada paciente han sido obtenidos a través de una red desarrollada por *Bitbrain* [12] y entrenada sobre el dataset *BOAS* [13]. Este conjunto de datos consta de 128 noches en las que los participantes fueron monitoreados simpultáneamente con dos tecnologías: un PSG de grado médico y la banda textil *Ikon Sleep*. Para garantizar una clasificación del sueño sólida y confiable, los datos fueron etiquetados de forma rigurosa. Tres expertos del sueño anotaron de forma independiente las grabacionees de PSG siguiendo los criterios de la *American Academy of Sleep Medicine* (*AASM*), y fue utilizado un cuarto como consenso. Al estar las grabaciones de ambas tecnologías perfectamente sincronizadas, estas etiquetas de fases del sueño pudieron aplicarse a los datos grabados con el sistema *Ikon Sleep*.

Finalmente, la red entrenada, con un nivel de concordancia del 86.64 % respecto a las etiquetas generadas por los expertos, se aplicó a los EEGs utilizados a lo largo de este trabajo dentro del cohorte HOGAR [14] para generar los hipnogramas.

Los hipnogramas se representan, por tanto, en una secuencia de números enteros, que codifican las fases del sueño de cada ventana de 30 segundos de la siguiente manera:

- 0 (W): despierto. Estado de vigilia con actividad cerebral elevada.
- 1 (N1): Fase No REM N1. Primera etapa del sueño ligero. Fase minoritaria de transición entre la vigilia y el sueño.
- 2 (N2): Fase No REM N2. Sueño ligero consolidado.
- 3 (N3): Fase No REM N3. Sueño profundo, importante para la recuperación física y consolidación de memoria.
- 4 (REM): Fase REM. Sueño con movimientos oculares rápidos, asociado a sueños vívidos y consolidación cognitiva.
- -1 ("Art"): Artefacto o pérdida de datos. La red no ha asignado una etiqueta válida a estas ventanas debido a ruido o datos insuficientes.

Se ha utilizado esta codificación ya que es perfecta para obtener determinadas features a raíz del hipnograma utilizando la librería *YASA*, además de ser la que establecen las guías y el criterio utilizado seguido por los expertos. En la Figura 5 podemos visualizar un hipnograma concreto, correspondiente a la sesión 1 del paciente 116, de más de 5 horas de duración.

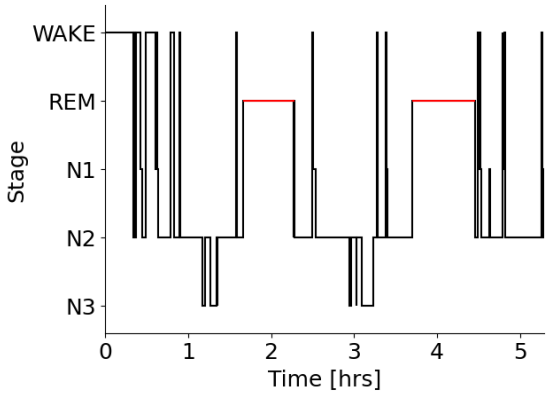


Figura 5: Ejemplo de hipnograma

Debido a que la longitud de las distintas sesiones es variable, se decidió eliminar todas aquellas de menos de 4 horas de duración, es decir, aquellas con menos de 480 ventanas de 30 segundos. Por otro lado, para aquellas de más de 9 horas, se han tenido en cuenta solamente las primeras 9 horas, es decir, las primeras 1080 ventanas de 30 segundos.

Además, de manera similar, las sesiones 1 de los pacientes 141 y 142 no llegan a fase N1, y la sesión 2 del paciente 136 no llega a la fase REM. Se ha decidido por tanto eliminar estas sesiones, puesto que la relevancia que pueden aportar 2 sesiones de 219 es considerada pequeña respecto al valor que pueden aportar las fases N1 y REM en el estudio. Más adelante veremos cómo se ha procedido de manera diferente respecto a las sesiones que no llegan a la fase N3.

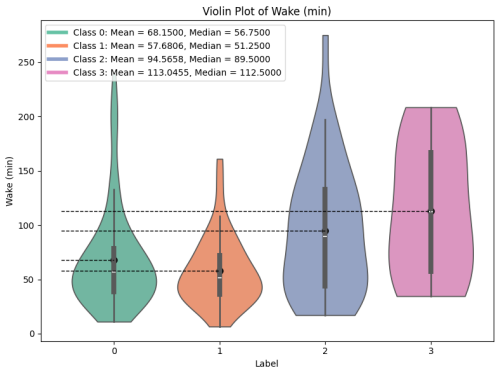
A partir de estos hipnogramas, el problema en este proyecto respecto a la macroestructura se ha planteado de dos formas distintas:

Evolución de las fases del sueño a lo largo del tiempo

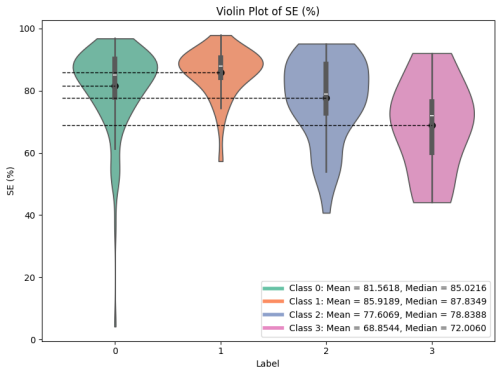
Fue necesario concatenar en un mismo fichero *.csv* todos los hipnogramas facilitados de todos los pacientes, a razón de uno por fila. De esta forma, cada sesión se representa como una *Serie Temporal Univariable Discreta (DUTS)*. Además, al final de cada una de las líneas de este nuevo fichero se añadió la etiqueta correspondiente al estado cognitivo de dicho paciente, presente en un fichero distinto. Aquellas sesiones correspondientes a pacientes cuyo estado cognitivo no estaba registrado, evidentemente, se descartaron.

Features derivadas del hipnograma

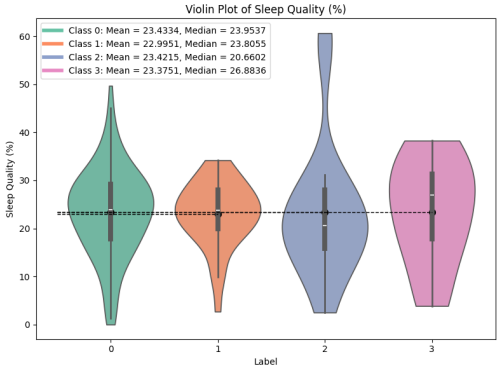
A raíz de la secuencia de números enteros que representan las fases del sueño del hipnograma, se obtuvieron 49 features, ligadas con las duraciones de las fases, las transiciones entre ellas o los distintos tiempos de latencia (tiempo hasta entrar en una fase), entre muchas otras. De esta forma, podemos caracterizar cada sesión como una serie de features obtenidas sobre la sesión completa. En la Figura 6 se puede observar la distribución de 3 de estas features en función del estado cognitivo de los pacientes. Es importante mencionar que, debido a que alrededor del 80 % de sesiones no llega a la fase N3, no se ha considerado la latencia a esta fase (i. e., el tiempo, en minutos, que tarda el paciente en llegar a N3). Toda esta lista de features se encuentra detallada en el Anexo A.



(a) Wake (min)



(b) SE (%)



(c) SQ (%)

Figura 6: *Violin plots* que muestran la distribución de algunas features derivadas de la macroestructura del sueño de los pacientes según su estado cognitivo. La Figura 6a muestra la duración total de la fase W. La Figura 6b visualiza el porcentaje de eficiencia del sueño. La Figura 6c muestra el porcentaje de calidad del sueño.

Sin embargo, no podemos olvidar que el uso de estas features como biomarcadores de la macroestructura del sueño hace que los datos pierdan la dimensión temporal con la que cuentan los hipnogramas implícitamente. Pasamos, por tanto, de una *DUTS* a una representación tabular, donde no se tiene en cuenta qué fases van antes o después de manera directa.

Estudio de la microestructura

La microestructura del sueño se refiere a los patrones breves y específicos de actividad cerebral que ocurren durante el sueño. Por ello, su análisis en este proyecto conlleva el tratamiento de señales EEG del dataset HOGAR, almacenadas en ficheros con formato *.npz*. Las dimensiones del objeto *NumPy array* de cada fichero son de la forma:

$$(\textit{número de ventanas}, \textit{frecuencia de sampleo} \times \textit{duración por ventana}, \textit{número de canales}),$$

siendo *número de ventanas* el número de ventanas totales de 30 segundos de la sesión (una sesión de 8 horas, por ejemplo, tendría 960 ventanas), *frecuencia de sampleo* la frecuencia a la que se

han decidido grabar los datos (en nuestro caso, tras un proceso de "downsampling" previo para reducir el tamaño del input de la red, se pasó de 256 a 128 Hz, como se explica más adelante), *duración por ventana* la duración de cada ventana de tiempo (30 segundos), y *número de canales* el número de canales utilizado (al usar el dispositivo *Ikon Sleep*, solamente 2). Al multiplicar la frecuencia de sampleo por la duración de una ventana tenemos el número de puntos temporales de cada ventana.

De esta forma, cada uno de estos ficheros *.npz* será un *NumPy array* que contendrá *número de ventanas* listas. Cada una de estas listas tendrá a su vez *frecuencia de sampleo*×*duración por ventana* ($128 \cdot 30 = 3840$) listas anidadas, una por cada valor medido. A su vez, cada una de estas listas contendrá 2 valores, uno por canal.

Las señales EEG analizadas durante este proyecto han sido previamente filtradas por investigadores de *Bitbrain* mediante diferentes algoritmos para eliminar artefactos. De esta forma, las señales se encuentran limpias para ser procesadas.

En primer lugar, se ha realizado un filtrado paso-banda entre 0.5 Hz y 45 Hz, de forma que se elimina la señal con valores por debajo y por encima de estos umbrales, respectivamente, ya que no contienen información relevante para los análisis que se desean llevar a cabo. En segundo lugar, se han detectado y eliminado artefactos en distintos rangos de estas frecuencias. Concretamente, se han eliminado aquellos correspondientes a ruido de alta frecuencia, entre 30 y 45 Hz y según un umbral preestablecido, producido por la tensión muscular o interferencias electromagnéticas. También se ha eliminado el ruido en baja frecuencia, entre 0.5 Hz y 4 Hz y según otro umbral preestablecido, causado principalmente por el sudor de la frente. Asimismo, se han eliminado segmentos de la señal planos debido a la pérdida de contacto entre los electrodos y la piel. Por último, se han eliminado aquellos artefactos correspondientes a amplitudes altas (< 2250 μV), como fallos temporales debido a los sensores o al movimiento de los cables que conectan los electrodos con los amplificadores. Todos los umbrales y rangos descritos han sido probados y utilizados habitualmente por investigadores de la empresa.

Feature engineering **de las señales cerebrales**

Las distintas bandas de frecuencia analizadas durante este trabajo vienen definidas por los siguientes rangos:

- **δ** (delta): 0.5 - 4 Hz
- **Θ** (theta): 4.1 - 8 Hz
- **α** (alpha): 8.1 - 13 Hz
- **β1** (beta 1): 13.1 - 15 Hz
- **β2** (beta 2): 15.1 - 22 Hz
- **β3** (beta 3): 22.1 - 30 Hz
- **γ** (gamma): 30.1 - 45 Hz

Analizar los EEGs en función de sus distintas bandas de frecuencia es fundamental, ya que cada banda está asociada con diferentes procesos cerebrales, estados cognitivos y funciones fisiológicas. Por ejemplo, en este estudio serán de especial interés las bandas δ, relacionada con el sueño profundo y procesos regenerativos [15], y α, que representa estados de calma [16] y en sueño

caracteriza la fase W. En la Figura 7 se pueden visualizar las formas que suelen tener las señales EEG en estas frecuencias.

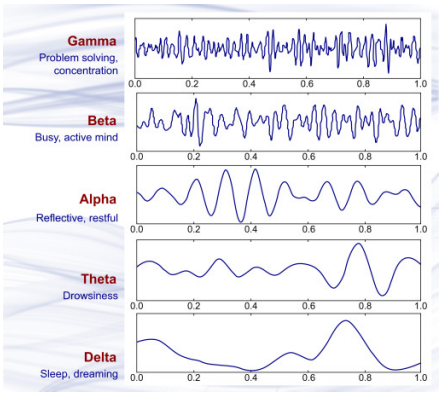


Figura 7: Visualización de distintas frecuencias de una señal EEG, a lo largo del tiempo [17].

Basándonos en el paper mencionado en la sección Estado del arte de este trabajo, se obtuvieron una serie de features de la microestructura del sueño, cuyas fórmulas se encuentran descritas en el Anexo A. Los hipnogramas son utilizados en los experimentos de esta sección para agrupar las features de la microestructura según la fase del sueño. Por ello, ya que muy pocas sesiones llegan a entrar en fase N3, no se utilizarán como features las que corresponderían a esta fase, debido al alto número de valores nulos para este tipo de features. A pesar de que modelos como los RFs (RFC y XGBRFC) son capaces de trabajar con valores nulos, otros (como los SVMs, LRs y DNNs) no lo son. Además, si se hubiera decidido simplemente asignar "0s" a estos valores nulos, los modelos estarían trabajando con información falsa.

Estas features de la microestructura se pueden dividir en 6 grupos, según su naturaleza:

Features en los dominios del tiempo y la frecuencia
En primer lugar, de cada ventana de 30 segundos se extrayeron features tanto en el dominio del tiempo como en el dominio de la frecuencia.

- **Dominio del tiempo**: *line length*, para tratar de medir la complejidad, y kurtosis, para describir la forma de la distribución de la señal completa. A pesar de que el paper también indica el cálculo de una feature conocida como "sample entropy" para cuantificar la complejidad de la señal, esta feature no se calculó debido a su altísimo tiempo de cómputo y a que esta feature no era reportada como una de las más discriminantes entre las clases. Por tanto, en el dominio del tiempo contamos con 2 features.
- **Dominio de la frecuencia**: ha sido necesario hacer uso del algoritmo de Welch. La versión utilizada, implementada por el departamento de *Data Science* de *Bitbrain*, calcula, entre otras métricas, la Densidad de la Potencia Espectral (*PSD - Power Spectral Density*) media dada una señal sampleada a una determinada frecuencia. Este algoritmo se ha utilizado con los siguientes parámetros:
  - **Tipo de ventana**: Hamming

- **Tamaño de ventana**: $2 \cdot fs = 2 \cdot 128 = 256$. Se ha elegido este factor "2" porque, como hemos comentado previamente, las señales a procesar se encuentran filtradas (filtrado paso banda) entre 0.5 Hz y 45 Hz. El factor se calcula como el periodo (T) correspondiente al límite inferior al que está filtrada dicha señal ($1/0.5 = 2$).
- ***Overlapping***: $1 \cdot fs = 1 \cdot 128 = 128$. Se ha elegido este factor "1"porque es común que el *overlapping* sea la mitad del tamaño total de la ventana, de manera que en cada paso el algoritmo se desplaza media ventana de tamaño.
- **nfft**: $10 \cdot fs = 1280$. Este número de puntos utilizado para calcular la FFT de cada bloque es adecuado para la resolución que necesitamos.
- **Frecuencia de sampleo (fs)**: 128 Hz.

Por tanto, utilizando este algoritmo de Welch con los parámetros descritos, se han calculado las siguientes features dentro del dominio de la frecuencia:

- **Potencia relativa de una banda**: representa la fracción de potencia contenida en cada banda de frecuencia respecto a la potencia total de la señal. Cada ventana de 30 segundos se dividió en sub-ventanas de 2 segundos. De esta forma, para cada una de las bandas de frecuencia analizadas en este trabajo (δ, Θ, α, β1, β2, β3 y γ), obtenemos la menor y mayor potencia relativa para cada ventana de 30 segundos (por ejemplo, obtenemos la menor potencia relativa de δ de todas las sub-ventanas de 2 segundos para cada ventana de 30 segundos). De la misma forma, calculamos también la potencia relativa media de cada banda para cada ventana de 30 segundos, así como la desviación típica. De esta forma obtenemos 28 features por cada ventana de 30 segundos.
- **Ratios de potencias relativas**: a pesar de que en este trabajo se analizan más bandas que en el paper de referencia (en este trabajo también se tienen en cuenta β1, β2, β3 y γ, además de δ, Θ y α), sólo se han calculado los ratios de las potencias relativas entre las bandas δ/Θ, δ/α y Θ/α, tal y como realiza el paper. De la misma forma que con las potencias relativas, cada ventana de 30 segundos se subdivide en sub-venatans de 2 segundos para quedarnos con el menor y el mayor valor de cada uno de estos ratios, así como sus correspondientes medias y desviaciones típicas. Calculamos por tanto, en este punto, 12 features por cada ventana de 30 segundos.
- **Kurtosis de bandas**: dividimos cada ventana de la misma forma que en las otras features de la micrestructura, obteniendo la kurtosis mínima, maxima y media, así como la desviación típica de cada una de las 7 bandas frecuenciales analizadas. Calculamos así 28 features para cada ventana de 30 segundos.

Frecuencias individuales del EEG y potencias de sub-bandas α
Estas features fueron extraídas únicamente de las ventana correspondientes a las fases W y N1. Las podemos subdividir en:

- ***TF* (Transition Frequency)**: mínima potencia en el rango de frecuencias de la banda Θ.
- ***IAF* (Individual Alpha Frequency)**: máxima potencia (mayor que el valor de TF) en el rango "extendido" de la frecuencia α (5-14 Hz).

Teniendo los rangos de cada una de las sub-bandas α para cada ventana de 30 segundos, computamos la *PSD* de las 3 sub-bandas, así como la PSD del ratio α_3 / α_2. Obtenemos por tanto 6

features correspondientes a esta sección.

Features de *spindles* y ondas lentas
Se han calculado diversos patrones relacionados con las *spindles* y ondas lentas en todas las ventana de 30 segundos correspondientes a la fase N2. Este tipo de actividad neurofisiológica tiene una fuerte vinculación con la consolidación de la memoria [18].

- ***Spindles***: se calcula el número total de spindles de cada canal en cada ventana de 30 segundos, así como su duración, amplitud y frecuencia medias a lo largo de dicha ventana. Además, para cada sesión completa se calcula el número total de *spindles* de cada canal.
- **Ondas lentas**: de manera similar, se calcula el número total de ondas lentas de cada canal en cada ventana de 30 segundos, así como su duración y frecuencia medias. También se computó el pico negativo y el *PTP* (tiempo entre picos de las ondas) medios. Por último, se calculó el número total de ondas lentas por canal a lo largo de toda la sesión.

Es importante recalcar que, en este caso, las features obtenidas para las ventanas de 30 segundos son promediadas respecto a aquellas ventanas que contaban con al menos un evento (*spindle* u onda lenta). Contamos con 11 features correspondientes a estos eventos.

En el proceso de obtención de estas features se detectó que existían usuarios que carecían de *spindles* para uno o ambos canales. Para que la media que se computaría posteriormente fuera lo más fiel a la realidad posible, se decidió que las features correspondientes a estas "missing *spindles*" fueran generadas artificialmente en base la media de las 10 sesiones más cercanas según la duración media de las ondas lentas de la sesión con "missing *spindles*" (según el algoritmo *kNN*, con $k = 10$). Si se hubiera decidido imputar estas features faltantes como ceros, se estaría generando información errónea (por ejemplo, una amplitud de 0 es una amplitud válida para realizar cálculos, pero falsa).

Al contrario que el paper de referencia [3], no se calcularon las features de coherencia entre canales, puesto que en este proyecto solo se utilizan los canales frontales. Estos, al encontrarse en el mismo área craneal, tienen una coherencia alta entre ellos, además de que no se disponen de otros canales con los que computar esta coherencia en el dispositivo utilizado.

Siguiendo la metodología del paper y utilizando los hipnogramas de cada sesión anteriormente descritos, todas estas features correspondientes a la microestructura fueron promediadas respecto a todas las ventanas de 30 segundos correspondientes a la misma fase del sueño. Debido a que más del 85 % de los participantes no alcanzaban la fase N3, esta fase no fue tenida en cuenta.

De manera similar, las ventanas caracterizadas como "Art" han sido también descartadas. Estas ventanas corresponden a señal de mala calidad identificadas por el detector de artefactos, es decir, podrían pertenecer a cualquier fase del sueño. Seguramente ni un experto ni la red utilizada seráin capaces de clasificarlas por el alto nivel de ruido. Por tanto, nos quedamos exclusivamente con las fases W, N1, N2 y REM.

Además, puesto que en el paper promedian las features en función de los canales próximos, promediamos las features obtenidas en nuestros 2 canales puesto que, al fin y al cabo, los 2 canales (*AF7* y *AF8*) se encuentran próximos en la zona frontal. Este promedio no se realiza en el caso de las features correspondientes a *spindles* y ondas lentas.

Grupo - Subgrupo	Feature	Nº de fases	Expresión	Total
T y F - T	*Line Length*	4	$4 \cdot 1$	4
	Kurtosis	4	$4 \cdot 1$	4
T y F - F	*RPs*	4	$4 \cdot 7 \cdot 4$	112
	RPs ratios	4	$4 \cdot 3 \cdot 4$	48
	Kurtosis de bandas	4	$4 \cdot 7 \cdot 4$	112
IF y PSD α_i - *IF*	*TF*	2	$2 \cdot 1$	2
	IAF	2	$2 \cdot 1$	2
IF y *PSD* α_i - *PSD* α_i	α_i *PSD*	2	$2 \cdot 4$	8
Spindles y ondas lentas	*Spindles*	1	$1 \cdot 5 \cdot 2$	10
	Ondas lentas	1	$1 \cdot 6 \cdot 2$	12
Total				**314**

Tabla 4: Desglose del total de features de la microestructura del sueño calculadas

Modelos utilizados

A lo largo de todo el proyecto se han probado diferentes modelos según la forma en la que se ha abordado el problema en cada momento.

Para tratar de clasificar el estado cognitivo a partir de la evolución de las fases del sueño a lo largo del tiempo se optó por el uso de Redes Neuronales Recurrentes (RNNs), puesto que es uno de los tipos de red más comunmente utilizado cuando el orden y la dependencia temporal entre elementos son importantes. Estas redes están diseñadas para capturar patrones a lo largo del tiempo. Concretamente, se experimentó con 2 arquitecturas de RNNs diferentes: *Long-Short Term Memory* (LSTMs) y *Gated Recurrent Units* (GRUs). El funcionamiento y las distintas arquitecturas de estas RNNs se describen en el Anexo B. Además, siguiendo una arquitectura multimodal propuesta para clasificar notas de estudiantes [19], se ha creado un modelo que admite como entrada tanto información secuencial (los hipnogramas) como información tabular acerca de la edad y el estado cognitivo del paciente. Este modelo alterna capas densas y capas LSTMs iterativamente, creando varias concatenaciones de estos tipos de capas. En la sección Evaluación se encuentran los distintos hiperparámetros probados de estos modelos.

Para la clasificación del estado cognitivo utilizando las features derivadas del hipnograma, así como para la clasificación utilizando también las features procedentes de la microestructura, se ha hecho uso de modelos de ML y DL que funcionan mejor con información estructurada de forma tabular. Los modelos seleccionados han sido lineales, como *Support Vector Machine* (SVM) con un *kernel* lineal y *Logistic Regression* (LR), y no lineales, como SVMs con *kernels* no lineales (*RBF* y polinómico de grado 3), *Random Forest Classifier* (RFC), *XGBoost Random Classifier* (XGBRFC) y 2 arquitecturas distintas de redes neuronales profundas (DNN1 y DNN2). De estas últimas, la primera cuenta con 4 capas ocultas (32, 32, 16 y 16 neuronas), y la segunda, con 3 capas ocultas (64, 64 y 32 neuronas). El funcionamiento, hiperparámetros y características de todos estos modelos están descritos en el Anexo B.

Para cada uno de estos modelos, se ha utilizado una función de pérdida (también llamada función objetivo), la cual el modelo correspondiente trata de minimizar a lo largo de su aprendizaje. De

esta forma, la función de pérdida "guía" el proceso de optimización al proporcionar una medida de qué tan bien (o mal) el modelo está realizando sus predicciones. Las funciones utilizadas han sido las que se encuentran por defecto en las librerías que implementan los diferentes modelos, siendo las más comunes en la literatura de problemas de clasificación. No obstante, al observarse que el uso de redes neuronales no era capaz de generalizar bien con datos desbalanceados (aún implementando la ponderación de pesos detalladas en el apartado "Problema del desequilibrio de clases" descrito a continuación), este tipo de modelos utilizó una función de pérdida distinta a la que viene por defecto en la librería utilizada.

Todas las funciones de pérdida utilizadas están descritas formalmente en el Anexo B.

Problema del desequilibrio de clases

Para lidiar con el gran desequilibrio entre las clases, se han ponderado los pesos durante el entrenamiento de todos los modelos (tanto de ML como de DL), de forma que las clases menos representadas adquieren pesos más altos, inversamente proporcionales a la frecuencia con la que aparecen en el conjunto de entrenamiento. La fórmula para calcular estos pesos es:

$$w_i = \frac{\text{Número total de sesiones}}{\text{Número total de sesiones de la clase } i},$$

siendo w_i el factor por el que se multiplicarán los pesos de cada clase en cada modelo.

Para que los modelos de DL fueran capaces de entrenar con estos nuevos pesos, fue necesario codificar las etiquetas siguiendo la estrategia *one-hot encoding*. De esta forma, tanto para los problemas de clasificación binaria (configuraciones de etiquetado 1 y 2) como para los de clasificación multiclase (configuración de etiquetado 3), las etiquetas se representan de la siguiente manera:

$$\mathbf{y} = \begin{cases} [1,0,0,0] & \text{si } y = 0\,(H) \\ [0,1,0,0] & \text{si } y = 1\,(SCI) \\ [0,0,1,0] & \text{si } y = 2\,(MCI) \\ [0,0,0,1] & \text{si } y = 3\,(DEM) \end{cases}$$

Además, esta codificación es utilizada en la función de activación *softmax*, la cual está presente en las neuronas de la última capa de todos los modelos de DL con los que se han realizado los experimentos de este proyecto. Su expresión matemática, así como su funcionamiento, se encuentra en el Anexo B.

Por otro lado, los modelos de DL han hecho uso de la función de pérdida *Focal Crossentropy Loss*, cuyo funcionamiento y parámetros se describen en el Anexo B. Concretamente, se ha utilizado una variante en función de un parámetro α. Esta función se enfoca en las predicciones en las que el modelo falla, en vez de aquellas en las que acierta [20], para asegurar que las predicciones en los sujetos (sesiones) con menor representación en el dataset mejoran a lo largo del tiempo. Se logra a través de un proceso denominado *Down Weighting*, que reduce la influencia de aquellos ejemplos "fáciles de predecir" (aquellos más frecuentes) para centrarse en los más difíciles de predecir. Esta función de pérdida es distinta a la *Entroía Cruzada Categórica*, la cual viene por defecto con la librería *Keras* cuando se utilizan modelos de DL para clasificación.

Evaluación

Experimentos

Todos los experimentos realizados en este proyecto se han llevado a cabo mediante un proceso de *k-fold Cross Validation*, con $k = 10$, por lo que cada conjunto de test de cada fold tendrá aproximadamente el 10 % del número total de datos del experimento. Debido a que existen pacientes con más de una sesión, fue necesario asegurarse de que en el momento de dividir los datos (tanto entre los subconjuntos de *train_validación* y *test* como en los subconjuntos de *train y validación* en los experimentos con modelos de DL) las sesiones correspondientes al mismo paciente estuvieran en el mismo conjunto. Por tanto, a pesar de que se le indicaba a la función que el 10 % de los datos debía ir al conjunto de test, no siempre era así (por ejemplo, a veces podía tener un 9 % ó un 11 %), puesto que prevalece el criterio de que las sesiones de un mismo paciente se encuentren en el mismo conjunto tras la división sobre el criterio de los porcentajes que caracterizan el tamaño de cada conjunto. Aseguramos así que el conjunto de test no haya sido visto previamente por el modelo en cada iteración de la CV, con el fin de evaluar correctamente las capacidades de generalización del mismo.

A la hora de dividir el resto de los datos en train y validación en el caso de experimentos con modelos de DL, la restricción de mantener todas las sesiones de un paciente en el mismo subconjunto se mantiene. Sin embargo, en este caso el número de sesiones en el conjunto de validación viene determinado por un parámetro, que representa la proporción del total (*train* y *validación*) correspondiente al subconjunto de *validación*. En todos los experimentos realizados, este parámetro será 0.2. Además, en cada fold las sesiones del correspondiente subconjunto de validación serán elegidos de manera aleatoria hasta completar el número de sesiones indicado.

A lo largo de todo el proyecto, el conjunto común de hiperparámetros probado para todos los modelos de DL descritos será el de la Tabla 5. Estos hiperparámetros se han ajustado sobre el subconjunto de validación, a medida que el modelo realiza su proceso de entrenamiento.

Hiperparámetro	**Valores**
Learning rate (*tasa de aprendizaje*)	{0.1, 0.01, 0.001, 0.0001, 0.00001}
Batch size (*tamaño del lote*)	{8, 16, 32, 64}
Algoritmos de optimización	{Adam, SGD, RMSProp}

Tabla 5: Hiperparámetros de los modelos de DL.

A pesar de que a la tasa de aprendizaje (*lr*) se le asigna un valor inicial en cada experimento, este valor se va ajustando dinámicamente a lo largo de las épocas de entrenamiento de un modelo. En todos los experimentos con modelos de DL llevados a cabo, por tanto, se ha seguido la estrategia de decaimiento por pasos, de forma que cada 100000 pasos esta tasa de aprendizaje se ajusta. El factor de decaimiento utilizado es de 0.96, por lo que la tasa de aprendizaje se multiplica por este valor cada vez que ocurre el ajuste. La fórmula general para este decaimiento por pasos se describe en el Anexo B.

Además de estos hiperparámetros, cada modelo será probado con una serie de hiperparámetros específicos de ese modelo. Este método de configurar hiperparámetros se realizará sobre el conjunto de validación en el caso de los modelos de DL utilizando la metodología *GridSearch* (GS),

que prueba todas las combinaciones de hiperparámetros posibles, siguiendo con las prácticas comunes. Esta estrategia hizo que los tiempos de computación aumentaran exponencialmente.

A la hora de elegir los mejores hiperparámetros, el conjunto de hiperparámetros elegido fue aquel cuya *validation loss* durante el entrenamiento fuera la más baja, en el caso de los modelos de DL. Además, debido a que todos los experimentos con modelos de DL han sido realizando una *10-fold CV*, el conjunto de hiperparámetros seleccionado finalmente ha sido aquel que más se ha repetido a lo largo de los folds. Si no había ningún conjunto "favorito" (ya sea por un empate entre distintos conjuntos o porque ninguno se ha repetido en más de un fold), se ha escogido aquel formado por el hiperparámetro i más repetido, junto al hiperparámetro j más repetido... y así sucesivamente. Si este era el caso, el conjunto de hiperparámetros debía reevaluarse. En el caso de los modelos de ML, el mejor conjunto de hiperparámetros simplemente era aquel cuya *accuracy* sobre el conjunto de test fuera la mejor, siguiendo la misma política que con los modelos de DL. Esto no solo aplica para esta sección, sino para todas las del proyecto que incluyan la descripción de los experimentos realizados.

Por otro lado, se han realizado distintos experimentos con arquitecturas RNN (descritas en el Anexo B), probando distintos números de capas ocultas y neuronas. En la Tabla 6 se considera "concatenación" la unión entre una capa LSTM y una capa densa (Figura B.4 del Anexo B). A pesar de denominarse como hiperparámetros en dicha tabla, es importante remarcar que cada uno de estos experimentos fue ejecutado de manera independiente, es decir, que no se seleccionaron estos hiperparámetros en función de un proceso de GS con CV.

Arquitectura	**Hiperparámetro**	**Valores**
Modelos con capas LSTM o GRU	Nº de neuronas en capas recurrentes	{16, 32, 64}
	Nº de capas recurrentes	{1, 3}
Modelo híbrido	Nº de neuronas en capas LSTM	{16, 32, 64}
	Nº de neuronas en capas densas	{16, 32, 64}
	Nº de concatenaciones	{1, 2, 3}

Tabla 6: Hiperparámetros probados en las arquitecturas RNN y el modelo híbrido.

En el entrenamiento de los modelos de DL no se ha utilizado el *callback EarlyStopping* (el cual detiene el entrenamiento si la *validation loss* comienza a subir, o si la *validation accuracy* no mejora durante un determinado número de épocas) debido a que se promedian los valores de las métricas de las curvas de aprendizaje a lo largo de todos los folds realizados. Por tanto, al utilizar estas curvas como feedback a lo largo de todos los experimentos realizados, todos los folds necesitan tener el mismo número de épocas para calcular las curvas de aprendizaje de una ejecución final, y saber así si el modelo aprende o no. Esto aplica para todos los modelos que utilicen redes neuronales a lo largo de todo este proyecto, independientemente de la sección en la que dichos modelos se encuentren.

El problema de este proyecto se ha abordado de 3 formas diferentes, a las que denominaremos "Approaches" a lo largo de todo el documento:

- ***Approach 1:*** Se ha utilizado la evolución de las fases del sueño a lo largo del tiempo (es decir, los hipnogramas en sí).
- ***Approach 2:*** Se han utilizado las features derivadas de los hipnogramas, obtenidas a partir de los hipnogramas originales.

- ***Approach 3:*** Se han combinado las features derivadas del hipnograma con aquellas derivadas de la microestructura. Además, para esta última forma de afrontar el problema, se ha seguido un proceso de "selección de features".

La selección de features de esta *Approach 3* ha constado de los siguientes pasos:

1. En primer lugar, de todas las features implementadas se seleccionaron aquellas que el paper reportaba como más discriminativas. Por un lado, se seleccionaron aquellas que diferenciaban entre sujetos sanos y con demencia (33 features en común con las 363 features implementadas), constituyendo el primer subconjunto de features, y también aquellas que diferenciaban entre sujetos sanos y con MCI (35 features), constituyendo el segundo subconjunto. Por otro lado, se realizó un test estadístico ANOVA para obtener las features más discriminantes para una configuración de etiquetado determinada, con un nivel de significancia (*p-valor*) de 0.05. En el caso de la configuración 1, el número de features discriminantes según este test es 43, mientras que para las otras dos configuraciones se ha limitado a las 50 con el p-valor más pequeño.

2. En segundo lugar, en cada experimento se entrenó el modelo correspondiente con uno de estos subconjuntos de features. Tras este entrenamiento se obtuvieron las 30 features con más relevancia según sus coeficientes de importancia. Estos coeficientes muestran qué valor asigna el modelo del experimento a cada feature a la hora de tomar la decisión de clasificar, tras ser entrenado. En el caso de las redes neuronales, estos coeficientes se tratan de valores SHAP, los cuales son estimados tras el entrenamiento. Los valores de estos coeficientes de importancia son promediados entre los 10 folds ejecutados en la CV. En los problemas de clasificación multiclase con los modelos SVM y LR, los coeficientes reportados por los modelos siguen una política "one-vs-one" (OvO). De esta manera, se devuelven los coeficientes más importantes para diferenciar todas las clases frente a todas. Por tanto, el número de conjuntos de coeficientes (N) vendrá dado por la expresión:

$$N = \binom{C}{2} = \frac{C(C-1)}{2},$$

   siendo C el número de clases. Por tanto, al contar con 4 clases, estos modelos reportarán 6 conjuntos de coeficientes. El conjunto elegido para las clasificaciones multiclase ha sido el que clasifica personas sanas (H) frente a personas con demencia (DEM).

3. En tercer lugar, se entrena el modelo correspondiente desde el principio con el nuevo subconjunto de features obtenido en el paso anterior gracias a los coeficientes de importancia.

Sin embargo, también se decidió no ejecutar el paso 1. El hecho de no utilizar tanto el test ANOVA (que selecciona aquellas features independientemente correlacionadas con la clase) como los subconjuntos de features reportadas por el paper hace que el modelo decida, sobre el conjunto total de features, cuáles son más importantes para dicho modelo directamente.

Al seleccionar las distintas features de esta manera se intenta solventar un problema típico conocido como la "maldición de la dimensionalidad" [21], que puede impactar negativamente en modelos de ML y DL. Al tener más features que muestras (en este último caso, 363 features frente a 219 sesiones), el modelo puede ajustarse demasiado a los datos de entrenamiento, captando ruido en lugar de patrones generalizables y, por tanto, fallando a la hora de clasificar nuevos datos.

La Figura 8 muestra una vista generalizada de la pipeline de estos experimentos.

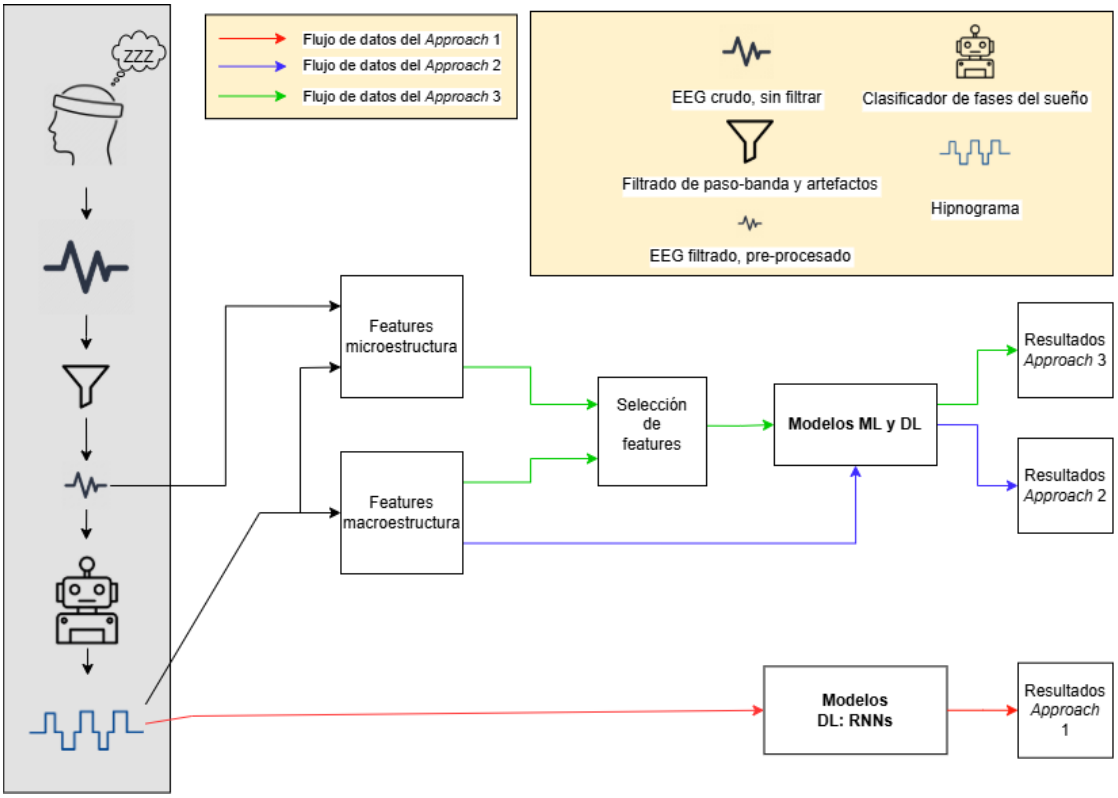


Figura 8: Pipeline del proyecto, que muestra los flujos de datos de las 3 formas de abordar el problema descritas, así como el preprocesado y la obtención de los datos.

El desglose del cálculo del número total de experimentos realizados en este proyecto se muestra en el Anexo B.

Métricas

Las métricas de evaluación de los diferentes modelos descritos han sido calculadas promediando sus valores entre los 10 folds correspondientes a la *CV*. Estas métricas son las siguientes:

- **Matriz de confusión**: matriz cuadrada de CxC dimensiones (siendo C el número de clases) que permite visualizar las predicciones del modelo y compararlas con las etiquetas reales en un problema de clasificación, lo que facilita la identificación de errores específicos que comete el modelo. Por tanto, obtendremos una matriz de dimensiones 2x2 para los experimentos realizados con las configuraciones de etiquetado 1 y 2, y de 4x4 para los realizados con la configuración 3. Debido a que todos los experimentos han sido realizados siguiendo una *k-fold Cross Validation* con $k = 10$, los valores en las celdas de estas matrices de confusión son porcentajes, y no el número natural de sujetos (en nuestro caso, sesiones) bien o mal clasificados. Estos porcentajes representan la media de los porcentajes de aciertos o errores a lo largo de las k iteraciones.
- ***Accuracy***: mide la proporción de predicciones correctas realizadas por un modelo en rela-

ción con el número total de muestras evaluadas. Por tanto, cuanto más cercano a 1 sea su valor, mejor clasificará una sesión de clase C como clase C. Podemos visualizar fácilmente el propósito de esta métrica de la siguiente manera:

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}}$$

Expresada de manera formal, la métrica sigue la siguiente fórmula matemática:

$$Accuracy = \frac{\sum_{i=1}^{n}(y_i = \hat{y}_i)}{n}$$

donde:

- $(y_i = \hat{y}_i)$ es una función indicadora que vale 1 si $y_i = \hat{y}_i$ y 0 en caso contrario.
- n es el número total de muestras.

Es una métrica básica y fácil de interpretar, aunque puede resultar engañosa cuando las clases están desbalanceadas, como es nuestro caso. Por ejemplo, el modelo puede obtener una alta *accuracy* simplemente prediciendo siempre la clase mayoritaria, ignorando las minoritarias.

Las siguientes métricas serán calculadas en función de *TP*, *FP*, *TN* y *FN*, donde:

- *TP*: número de verdaderos positivos, es decir, aquellos que se han clasificado como clase positiva i correspondientes a la clase positiva i.
- *FP*: número de falsos positivos, es decir, aquellos que se han clasificado como pertenecientes a la clase positiva i no correspondientes a la clase positiva i.
- *TN*: número de verdaderos negativos, es decir, aquellos que se han clasificado como clase negativa i correspondientes a la clase negativa i.
- *FN*: número de falsos negativos, es decir, aquellos que se han clasificado como clase negativa i no correspondientes a la clase negativa i.

- ***Balanced accuracy***: mide la media aritmética de la sensibilidad (o *Recall*, definida más tarde en esta sección) y la especificidad (*TN / (TN + FP)*) para una clase. Por tanto, para las situaciones con una clasificación binaria podemos definirla de la siguiente forma:

$$Balanced\ accuracy = \frac{sensibilidad + especificidad}{2}$$

Para los escenarios con una clasificación multiclase, la *balanced accuracy* corresponde a la métrica *Recall macro*, definida también en esta sección.

A diferencia de la *accuracy* estándar, la *balanced accuracy* evalúa el desempeño del modelo considerando por igual el peso de todas las clases, independientemente de su frecuencia.

- ***Precision***: mide la proporción de ejemplos correctamente clasificados como positivos entre todos los ejemplos que el modelo predijo como positivos. Es especialmente relevante cuando

el costo de los falsos positivos es alto, ya que mide la confianza del modelo al etiquetar una muestra como positiva. La métrica sigue la siguiente fórmula:

$$Precision = \frac{TP}{TP + FP}$$

Esta fórmula es adecuada para aquellos casos en los que solo tuviéramos 2 clases, y además una clase la consideráramos como "positiva". Sin embargo, debido a que la configuración 3 cuenta con 4 clases (personas sanas, con SCI, con MCI y con demencia), y además tratamos ninguna clase en especial como la clase "positiva" en las configuraciones 1 y 2, el cálculo de la precision debe enfocarse desde un problema de clasificación multiclase. En vez de calcular la *precision* por clase, lo que resultaría en 4 métricas distintas, se ha decidido calcular la *precision* promedio "macro" (macro-averaged precision). De esta manera, se calcula la *precision* por clase, obteniendo 4 métricas diferentes, para luego promediar los valores. Esta es calculada de la siguiente manera:

$$Precision\ Macro = \frac{1}{C} \sum_{i=1}^{C} Precision_i \tag{1}$$

donde C es el número total de clases (en nuestro caso, 2 ó 4), y Precision_i la precision para la clase i. De esta manera, ponderamos todas las clases por igual. Además, esta forma de calcular la precisión es útil y representativa cuando las clases están desbalanceadas, como es nuestro caso.

- ***Recall***: también conocida como sensibilidad, o tasa de verdaderos positivos, mide la capacidad de un modelo para identificar correctamente todas las instancias positivas de una clase. En otras palabras, indica cuántos de los elementos realmente positivos han sido correctamente clasificados como positivos por el modelo. Sigue la siguiente fórmula:

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

De manera similar a la precisión, el enfoque tomado para calcular el recall en la configuración de etiquetado 3 se calcula realizando el promedio "macro" mediante la siguiente fórmula:

$$Recall\ Macro = \frac{1}{C} \sum_{i=1}^{C} Recall_i \tag{3}$$

- ***F1-score***: combina la precisión y el recall en un solo valor, proporcionando una medida balanceada de la exactitud del modelo cuando hay un desbalance en las clases al buscar un equilibrio entre los verdaderos positivos, falsos positivos y falsos negativos. Se calcula:

$$F1\text{-}Score = 2 \cdot \frac{Precisión \cdot Recall}{Precisión + Recall} \tag{4}$$

Al igual que con la *precision* y el *recall*, calculamos el *F1-score* para la configuración de etiquetado 3 de la siguiente forma:

$$F1\text{-}Score\ Macro = \frac{1}{C} \sum_{i=1}^{C} F1\text{-}score_i \tag{5}$$

- ***Test loss***: valor de la función de pérdida calculado en el conjunto de datos de prueba. Refleja la "diferencia" entre las predicciones del modelo y los valores verdaderos de las etiquetas en el conjunto de datos de prueba. No se reporta en los resultados, pero ha sido útil a lo largo de la realización de los experimentos con DNNs para monitorizar el rendimiento de estos modelos.

Resultados y discusión

En esta sección se expondrán los resultados más relevantes, así como aquellos que reportan mejores métricas de todos los experimentos realizados.

Approach 1: Evolución de las fases del sueño a lo largo del tiempo

En los experimentos realizados tratando el problema como el análisis de la evolución de las fases del sueño, los resultados no han sido tan favorables como se esperaba. Las arquitecturas RNN probadas no han sido capaces de detectar apropiadamente patrones específicos y claros en estas secuencias, generadas por un modelo externo a partir de los EEGs. Los resultados para las distintas configuraciones son similares, si bien destaca con unos valores ligeramente mayores la red LSTM4, con 3 capas recurrentes y 16 neuronas en cada una de ellas, que clasifica entre sesiones correspondientes a pacientes sanos y con SCI respecto a aquellos con MCI y demencia. En la Figura 9 se muestran los resultados correspondientes a este experimento.

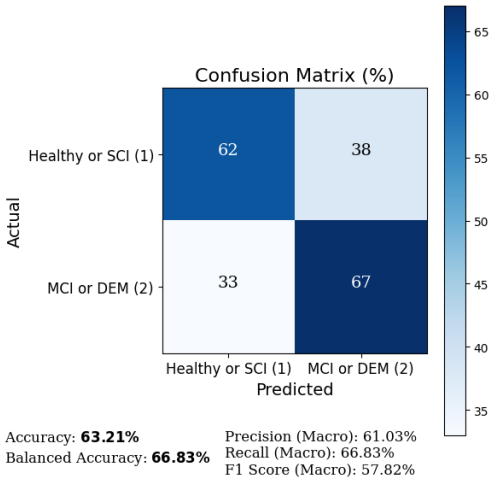


Figura 9: Matriz de confusión resultante de utilizar el modelo LSTM4, con una capa LSTM, bs=32, lr=0.0001 y entrenada durante 200 épocas, para la configuración de etiquetado 2.

Approach 2: Features derivadas de los hipnogramas

En los experimentos llevados a cabo con las features derivadas de los hipnogramas, es decir, sin dimensión temporal, los resultados tampoco han sido esclarecedores. El modelo que mejores

métricas reportó corresponde al modelo de red neuronal DNN1, de nuevo para la configuración de etiquetado 2. Su matriz de confusion y métricas se muestran en la Figura 10:

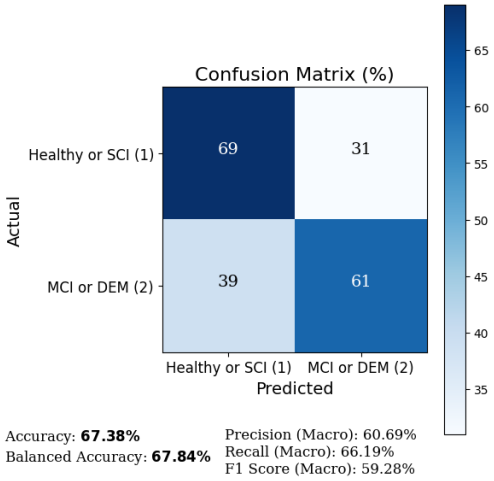


Figura 10: Matriz de confusión resultante de utilizar el modelo DNN1, con bs=8, lr=0.01 y entrenada durante 200 épocas, para la configuración de etiquetado 2.

Debido a que los modelos utilizados no tuvieron el rendimiento esperado, en este punto se estudió de forma paralela cómo influían otras variables, como diversos tests cognitivos (entre los que destaca el test *MoCA*) o la edad, en el aprendizaje del "mejor" modelo probado en esta sección. Las diferencias en los resultados al incluir este tipo de variables se pueden apreciar en el Anexo D. El hecho de conseguir un buen rendimiento con una cantidad de tests inferior a la que necesitan los expertos significa también un avance en este campo de investigación. En el futuro, este avance sería de utilidad para el desarrollo de una herramienta de soporte para el diagnóstico precoz.

Approach 3: Combinación de microestructura y macroestructura

Como ya se ha mencionado anteriormente, los resultados procedentes de combinar features obtenidas a partir de la microestructura del sueño con aquellas derivadas de la macroestructura han sido los más prometedores. En concreto, el modelo que mejor rendimiento ha conseguido ha sido el SVM (con *kernel* lineal), reportando un 87.68 % para la configuración 2 y un 73.53 % para la configuración 1. Estos resultados han sido alcanzados sin hacer uso del test estadístico ANOVA, por lo que se han utilizado los coeficientes de importancia de las features de manera directa, como se ha detallado previamente. Siguiendo esta metodología, y como era esperable, los resultados de este modelo para clasificar las 4 clases por separado (configuración 3) es mucho

menor, consiguiendo aún así un 47.47 %, muy por encima del azar. Los resultados de este modelo SVM se observan en la Figura 11.

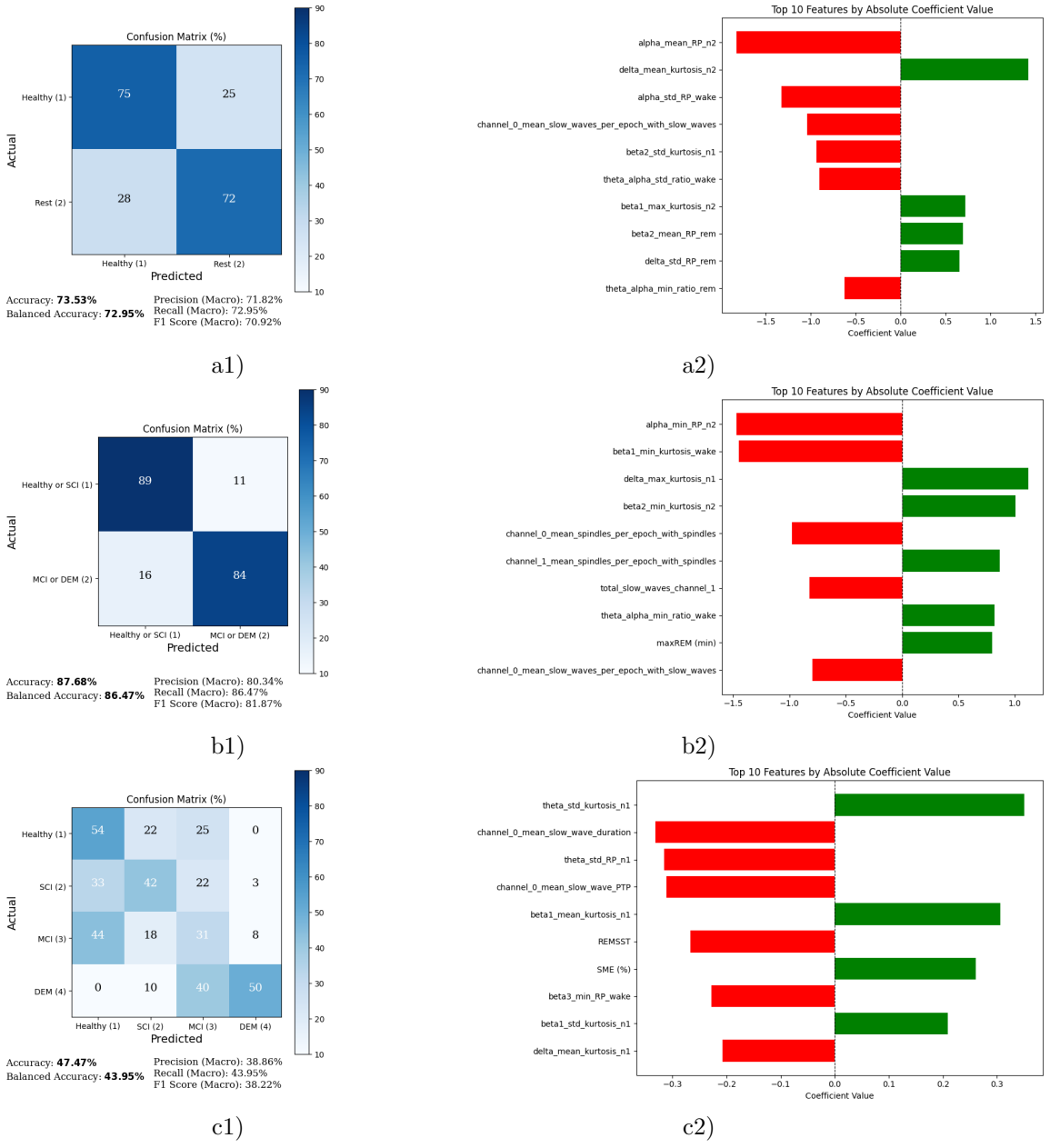
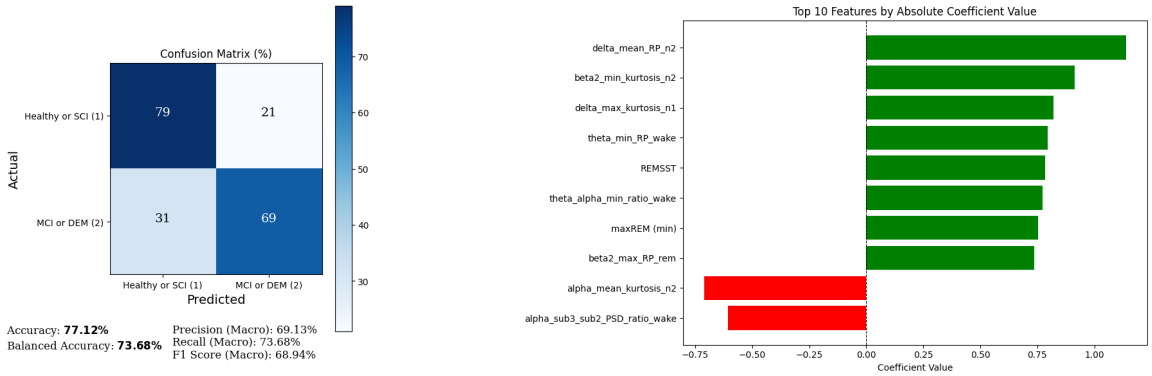


Figura 11: Resultados del modelo SVM para las 3 configuraciones de etiquetado. Se muestran también los coeficientes de importancia asignados (subfiguras a2), b2) y c2) de esta Figura 11) para las 10 features más importantes según el modelo SVM.

Además, hay que recalcar que debido a que, de las 3 formas de preseleccionar las features (antes

de seleccionarlas mediante coeficientes), el test ANOVA era el que mejores resultados reportaba en reglas generales, se han descartado las otras dos formas de proceder con la preselección, que utilizaban las features reportadas como importantes por el paper de referencia.

Los resultados reportados por el modelo de Regresión Logística (LR) también fueron bastante favorables, consiguiendo su mejor rendimiento en la configuración 2, con una *accuracy* del 77.12 % sin utilizar ANOVA para la preselección de features. El resultado correspondiente a este modelo y configuración se muestra en la Figura 12.

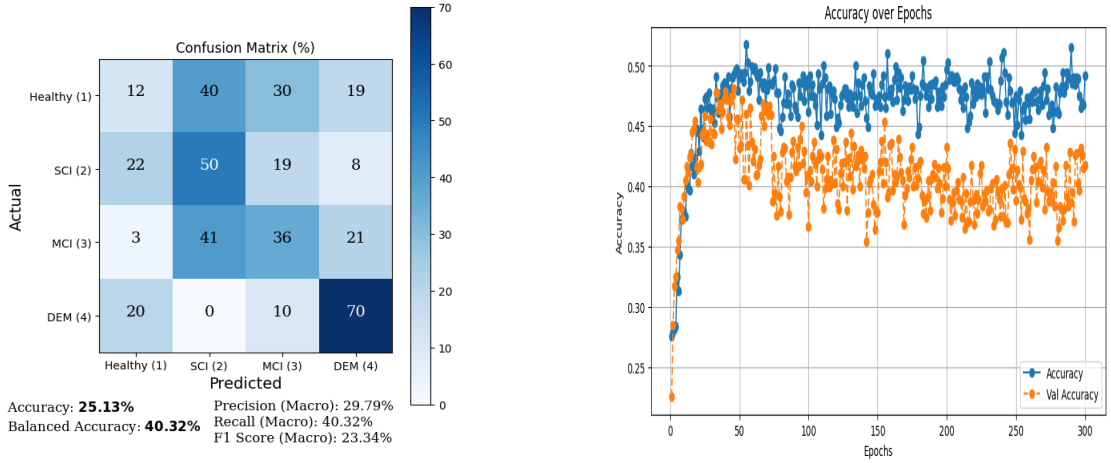


a) b)

Figura 12: Resultados del mejor experimento con el modelo LR combinando features de la microestructura y de la macroestructura. La subfigura a) muestra las métricas, y la b) las features más discriminativas, con sus coeficientes de importancia.

Por otro lado, los modelos de Random Forest (Random Forest Classifier y XGBoost Random Forest Classifier) entrenados han obtenido métricas considerablemente peores para las 3 configuraciones de etiquetado respecto a los demás modelos de ML. Por ejemplo, este modelo reporta una *accuracy* del 83.61 % para la configuración 2. Sin embargo, esta se debe principalmente a la gran precisión con la que ha clasificado los pacientes sanos como tal (98 %). Por el contrario, el número de falsos negativos para este modelo ha sido del 67 %, siendo además este un claro ejemplo de cómo la *accuracy* puede no ser una buena métrica en determinados escenarios.

Por último, los modelos que han sido capaces de clasificar mejor la clase DEM entre las 4 clases han sido las redes neuronales, a pesar de que su rendimiento tanto para las demás clases como para las clasificaciones binarias no ha sido tan alto como los modelos anteriormente descritos. La mejor arquitectura probada, haciendo uso de ANOVA para preseleccionar las features, ha conseguido una *balanced accuracy* del 40.32 % (*accuracy* del 25.13 % debido principalmente a su baja precisión para sesiones de pacientes sanos), clasificando el 70 % de las sesiones correspondientes a pacientes con demencia como tal (Figura 13 a)). Principalmente, esto se debe al ajuste realizado con la función de pérdida Focal Loss, que se centra más en aquellas sesiones más complicadas de clasificar (que, en nuestro caso, son las de demencia, pues el modelo ha tenido un número mucho menor de sesiones para ser entrenado respecto a las sesiones de pacientes sanos). Además, la misma arquitectura de red también reportó una *accuracy* del 65.28 % para la configuración 1.



a) b)

Figura 13: a) Matriz de confusión resultante de utilizar el modelo DNN2, con *bs*=8, *lr=0.001* y entrenada durante 200 épocas, para la configuración de etiquetado 3. b) Curvas de aprendizaje (en función de los valores de *accuracy* en los conjuntos de *train* y validación), a lo largo de las épocas de entrenamiento. La curva azul representa cómo el modelo es capaz de aprender los datos con los que está siendo entrenado. La naranja representa cómo el modelo es capaz de ir clasificando nuevas sesiones a lo largo de dicho entrenamiento.

Asimismo, una arquitectura con una capa más, pero menos neuronas por capa, reportó una *accuracy* del 70.80 %, de nuevo para la configuración 2, y del 34.68 % para la configuración 3.

En las Tablas 7 y 8 se pueden apreciar cómo el modelo SVM destaca entre los modelos probados, cuando se ha llevado a cabo la selección de features sin incluir el paso del test ANOVA. Asimismo, también se observa cómo al entrenar los modelos diviendo los sujetos en sanos y con SCI por un lado y con MCI y DEM por otro se consiguen los mejores resultados en líneas generales, al diferenciar entre aquellos sin patología diagnosticada y con patología. Las arquitecturas de las redes neuronales denominadas en la tabla como DNN1 y DNN2 se detallan en el Anexo B. En ambas tablas, así como en aquellas en el Anexo C, sólo se ha reportado los resultados del modelo SVM utilizando un kernel lineal, debido a que el rendimiento con los otros 2 kernels (polinómico de grado 3 y *RBF*) eran muy similares. De la misma manera, respecto a los modelos basados en Random Forest (RFC y XGBRFC) sólo se han reportado los resultados procedentes de los experimentos utilizando el criterio de pureza Gini.

Modelo	H vs SCI, MCI y DEM	H y SCI vs MCI y DEM	4 clases
SVM	73.53	87.68	47.47
LR	62.08	77.12	46.97
RFC	73.07	83.01	66.67
XGBRFC	70.71	84.03	68.96
DNN1	55.63	66.19	38.35
DNN2	69.89	70.37	44.22

Tabla 7: *Test accuracies* (%) de los modelos probados combinando micro y macroestructura, sin utilizar ANOVA para la preselección de features

Modelo	H vs SCI, MCI y DEM	H y SCI vs MCI y DEM	4 clases
SVM	72.95	86.47	43.95
LR	61.17	73.68	47.50
RFC	69.09	65.87	38.38
XGBRFC	66.30	70.04	47.67
DNN1	52.45	67.45	38.83
DNN2	66.46	64.89	48.80

Tabla 8: *Balanced test accuracies* (%) de los modelos probados combinando micro y macroestructura, sin utilizar ANOVA para la preselección de features.

Estas dos tablas muestran cómo las métricas correspondientes a los *Random Forest Classifiers* cambian mucho en función de si se toma en cuenta el número de sesiones de cada clase (*balanced accuracy*) o no (*accuracy*). Por ejemplo se observa que, a pesar de que para la configuración 2 el modelo RFC consigue un 83.01 % de *accuracy*, su *balanced accuracy* desciende hasta 65.87 %, debido a que el modelo ha clasificado el 96 % de las sesiones de pacientes sanos (muchas más que las otras clases) como pertenecientes a pacientes sanos, justo al contrario de como sucede con el mejor experimento reportado por la red DNN2.

Los resultados más relevantes haciendo uso de la preselección de features mediante ANOVA antes de escoger las más importantes utilizando los coeficientes están presentes en el Anexo C.

Features más discriminativas

Los valores de features de las potencias relativas de la banda α (y δ, para el modelo LR con la configuración 2) durante la fase mayoritaria N2 sí llevan a que el modelo decida si la sesión corresponde a una persona con un estado cognitivo más saludable (sano o SCI) o con un deterioro cognitivo más avanzado. Otras features, como la kurtosis de las bandas δ (muy presentes durante el sueño), $\beta 1$ y $\beta 2$ también parece que ayuden tanto al SVM como a la LR a clasificar las sesiones de forma binaria, siguiendo las configuraciones 1 y 2. La media (en cada ventana de 30 segundos) de ondas lentas que, junto a las *spindles*, están relacionadas directamente con la consolidación de la memoria también adquieren valores negativos altos en sus coeficientes de importancia, situándose como la $4^{\underline{a}}$ feature más importante en el SVM con la configuración 1 y como la $10^{\underline{a}}$ con la configuración 2 del mismo modelo. Respecto a las ondas lentas para el mismo modelo y la configuración 3, se observa que la duración de dichas ondas, así como la media de tiempo entre picos de las mismas, indican también que el modelo puede tomarlas como importantes a la hora de clasificar. De la misma manera, la media de *spindles* en ambos canales también son consideradas por el modelo que trata la configuración 2 como las $5^{\underline{a}}$ y $6^{\underline{a}}$ features más importantes. Sin embargo, y al contrario de lo que se pensaría en un principio, la media de *spindles* por ventana para el canal 0 tiene un coeficiente de importancia negativo, mientras que la del canal 1 tiene prácticamente el mismo valor, pero positivo. Este descubrimiento fue ciertamente sorprendente, pues ambos canales deberían reportar patrones de *spindles* similares al situarse muy próximos entre sí y recoger información del mismo área cerebral. Por otro lado, y como se esperaba, ninguna feature relacionada con las ondas γ ha sido reportada como discriminativa por ninguno de los modelos con mejores métricas. Por último, a pesar de que se puede observar fácilmente que las features correspondientes a la macroestructura no están tan presentes en las listas de features más discriminativas, aquellas que aparecen están mayoritariamente relacionadas con el sueño de la fase REM (no debemos olvidar que las features de la fase N3 no se han tenido en cuenta en este proyecto). Tanto la duración máxima consecutiva de los periodos en fase REM (modelos SVM y LR con la configuración 2) como el número de transiciones a la fase REM (SVM para la configuración 3 y LR para la configuración 2) adquieren altos valores para sus coeficientes de importancia.

Conclusiones y trabajo futuro

Los modelos utilizados para realizar los múltiples experimentos muestran ciertas correlaciones entre diferentes variables obtenidas a partir del análisis del sueño y el estado cognitivo de personas mayores. Varias features relacionadas con la consolidación de la memoria, la plasticidad neuronal y el mantenimiento general de la salud cerebral son reportadas en este trabajo como discriminativas a la hora de determinar el estado cognitivo de una persona. Además, en general los mejores resultados se han conseguido en la configuración de etiquetado 2, lo cual tiene sentido: al fin y al cabo, esta configuración divide los sujetos entre aquellos sin patología diagnosticada (H y SCI) frente a aquellos con una patología (MCI y DEM).

Los pocos estudios en la literatura científica que abordan en cierta manera el tema de este proyecto utilizan datos obtenidos con PSGs, con más canales y, lo más importante, muchos más participantes. Aún así, este proyecto ha cumplido con los objetivos iniciales propuestos, puesto que se ha explorado la relación entre la forma en la que las personas mayores duermen y su implicación directa en el estado cognitivo.

Sin embargo, a pesar de que este trabajo puede servir como una de las bases a la hora de utilizar métodos de inteligencia artificial para detectar el deterioro cognitivo a través de los EEGs durante el sueño debido a las métricas reportadas para diversos modelos, todavía queda mucho por hacer. La combinación de estos 3 elementos es un campo poco estudiado en la literatura científica, por lo que existe un gran número de métodos y modelos con distintas arquitecturas, tanto de ML como de DL, para mejorar los resultados expuestos en este trabajo. Desde aquí, se propone el uso de arquitecturas con mecanismos de atención (Time Series Transformer o Temporal Fusion Transformer) para tratar de encontrar patrones en las secuencias de fases del sueño que caracterizan a los hipnogramas, así como modelos de redes convolucionales de una dimensión (*1D-CNN*), arquitecturas especializadas (*Rocket*) o cualquier combinación de estos modelos. También se sugiere analizar la microestructura de forma diferente a la de este trabajo, no sólo obteniendo nuevas features de los EEGs, sino también realizando distintos filtrados y métodos de preprocesamiento. Por supuesto, se podrían probar nuevos modelos no descritos aquí, como modelos especializados para tratar EEGs (*EEGNet, TSFresh...*), para después tratar de combinarlos entre sí de forma coherente. Asimismo, sería interesante no contar solo con información procedente de las señales EEG, sino también con información sociodemográfica, como el género, la raza o el nivel educacional, para observar cómo afecta al estado cognitivo y en qué medida ayuda a su clasificación. Por último, y como ya se ha comentado, realizar estudios con más participantes, con un número más equilibrado entre los diferentes estados cognitivos, así como utilizar dispositivos médicos como PSGs sería interesante para analizar en qué medida la forma en la que una persona duerme permite decodificar su estado cognitivo.

Bibliografía

World Health Organization. (2023). Dementia. https://www.who.int/news-room/fact-sheets/detail/dementia#:~:text=Alzheimer%20disease%20is%20the%20most%20common%20form%20and%20may%20contribute,frontal%20lobe%20of%20the%20brain).

Lam, A. K. F., Carrick, J., Kao, C.-H., Phillips, C. L., Zheng, Y. Z., Yee, B. J., Kim, J. W., Grunstein, R. R., Naismith, S. L., & D'Rozario, A. L. (2024). Electroencephalographic slowing during REM sleep in older adults with subjective cognitive impairment and mild cognitive impairment. *Sleep*, *47*, zsae051. https://doi.org/10.1093/sleep/zsae051

Ye, E. M., Sun, H., Krishnamurthy, P. V., Adra, N., Ganglberger, W., Thomas, R. J., Lam, A. D., & Westover, M. B. (2023). Dementia detection from brain activity during sleep. *Sleep*, *46*, zsac286. https://doi.org/10.1093/sleep/zsac286

Danielle Pacheco, D. A. S. (2023). How Much Does A Sleep Study Cost? https://www.sleepfoundation.org/sleep-studies/how-much-does-a-sleep-study-cost

D'Atri, A., Scarpelli, S., Gorgoni, M., Truglia, I., Lauri, G., Cordone, S., Ferrara, M., Marra, C., Rossini, P. M., & De Gennaro, L. (2021). EEG alterations during wake and sleep in mild cognitive impairment and Alzheimer's disease. *iScience*, *24*, 102386. https://doi.org/10.1016/j.isci.2021.102386

Geng, D., Wang, C., Fu, Z., Zhang, Y., Yang, K., & An, H. (2022). Sleep EEG-Based Approach to Detect Mild Cognitive Impairment. *Frontiers in Aging Neuroscience*, *14*, 865558. https://doi.org/10.3389/fnagi.2022.865558

Chatur, A., Haghi, M., Ganapathy, N., TaheriNejad, N., Seepold, R., & Madrid, N. M. (2024). Advanced Classifiers and Feature Reduction for Accurate Insomnia Detection Using Multimodal Dataset. *IEEE Access*, *12*, 150664-150678. https://doi.org/10.1109/ACCESS.2024.3456904

Acharya, M., Deo, R. C., Tao, X., Barua, P. D., Devi, A., Atmakuru, A., & Tan, R.-S. (2025). Deep learning techniques for automated Alzheimer's and mild cognitive impairment disease using EEG signals: A comprehensive review of the last decade (2013 - 2024). *Computer Methods and Programs in Biomedicine*, *259*, 108506. https://doi.org/https://doi.org/10.1016/j.cmpb.2024.108506

Meghdadi, A., Levendowski, D., Kovacevic, N., Hamilton, J., Boeve, B., St Louis, E., Salat, D., & Berka, C. (2023). Comparison of sleep and wake EEG biomarkers in mild cognitive impairment and Alzheimer's disease. *Alzheimer's Dementia*, *19*. https://doi.org/10.1002/alz.075357

Hugo, J., & Ganguli, M. (2014). Dementia and cognitive impairment: epidemiology, diagnosis, and treatment. *Clinics in Geriatric Medicine*, *30*, 421-442. https://doi.org/10.1016/j.cger.2014.04.001

Sinai, C. (2024). Subjective Cognitive Impairment (SCI). https://www.cedars-sinai.org/health-library/diseases-and-conditions/s/subjective-cognitive-impairment-sci.html#:~:text=Overview,be%20verified%20by%20standard%20tests.

Esparza-Iaizzo, M., Sierra-Torralba, M., Klinzing, J. G., Minguez, J., Montesano, L., & López-Larraz, E. (2024). Automatic sleep scoring for real-time monitoring and stimulation in individuals with and without sleep apnea. *bioRxiv*. https://doi.org/10.1101/2024.06.12.597764

López-Larraz, E., Sierra-Torralba, M., Clemente, S., Fierro, G., Oriol, D., Minguez, J., Montesano, L., & Klinzing, J. G. (2024). *"Bitbrain Open Access Sleep Dataset"*. OpenNeuro. https://doi.org/doi:10.18112/openneuro.ds005555.v1.0.0

López-Larraz, E., Robledo-Menéndez, A., Jubera-Garcia, E., López-López, A., Simón-Lobera, P., Gelonch, O., De Francisco Moure, J., Marín, J. M., Molina-Torres, N., Osta, R., Lobo,

E., Lobo, A., Modrego, P. J., Magallón-Botaya, R., & Minguez, J. (2024). The HOGAR study: Home-based brain monitoring with a self-managed EEG to study cognitive decline in the ageing population. *17th Clinical Trials on Alzheimer's Disease (CTAD) Conference*.

BioSource Faculty. (2024). A Deep Dive into the Delta Rhythm. https://www.biosourcesoftware.com/post/a-deep-dive-into-the-delta-rhythm#:~:text=Delta%20waves%20are%20a%20hallmark,synaptic%20pruning%2C%20and%20memory%20consolidation.

Kendra Cherry. (2023). What Are Alpha Brain Waves? Increasing alpha waves may reduce depression. https://www.verywellmind.com/what-are-alpha-brain-waves-5113721

Priyanka A. Abhang, S. C. M., Bharti W. Gawali. (2016). Introduction to EEG-and Speech-Based Emotion Recognition. *Science Direct*, 19-50. https://doi.org/https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/brain-waves

Wei, Y., Krishnan, G. P., Komarov, M., & Bazhenov, M. (2018). Differential roles of sleep spindles and sleep slow oscillations in memory consolidation. *PLoS Computational Biology*, *14*, e1006322. https://doi.org/10.1371/journal.pcbi.1006322

Prabowo, H., Hidayat, A. A., Cenggoro, T. W., Rahutomo, R., Purwandari, K., & Pardamean, B. (2021). Aggregating Time Series and Tabular Data in Deep Learning Model for University Students' GPA Prediction. *IEEE Access*, *9*, 87370-87377. https://doi.org/10.1109/ACCESS.2021.3088152

Roshan Nayak. (2022). Focal Loss: A better alternative for Cross-Entropy. https://towardsdatascience.com/focal-loss-a-better-alternative-for-cross-entropy-1d073d92d075

Venkat, N. (2018). The Curse of Dimensionality: Inside Out. https://doi.org/10.13140/RG.2.2.29631.36006

Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*, 70-73. https://doi.org/10.1109/TAU.1967.1161901

Amanatullah. (2023). Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions. https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54

Wikipedia. (2024). Hinge loss. https://en.wikipedia.org/w/index.php?title=Hinge_loss&oldid=1239477748

Nagpal, A. (2017). L1 and L2 Regularization Methods. https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

Ruder, S. (2017). An overview of gradient descent optimization algorithms. https://arxiv.org/abs/1609.04747

Yadav, H. (2022). Dropout in Neural Networks. https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9

Grober, E., Sanders, A. E., Hall, C., & Lipton, R. B. (2010). Free and cued selective reminding identifies very mild dementia in primary care. *Alzheimer Disease and Associated Disorders*, *24*, 284-290. https://doi.org/10.1097/WAD.0b013e3181cfc78b

Perez-Valero, E., Lopez-Gordo, M. Á., Gutiérrez, C. M., Carrera-Muñoz, I., & Vílchez-Carrillo, R. M. (2022). A self-driven approach for multi-class discrimination in Alzheimer's disease based on wearable EEG. *Computer Methods and Programs in Biomedicine*, *220*, 106841. https://doi.org/https://doi.org/10.1016/j.cmpb.2022.106841

Martin-Loeches, M., Garcia-Trapero, J., Gil, P., & Rubia, F. J. (1991). Topography of mobility and complexity parameters of the EEG in Alzheimer's disease. *Biological Psychiatry*, *30*, 1111-1121. https://doi.org/10.1016/0006-3223(91)90181-k

Agradecimientos

Me gustaría agradecer principalmente a mis directores de TFG Eduardo López-Larraz y María Sierra Torralba, así como a Laura Pampliega, por haberme guiado y ayudado a lo largo de la consecución de esta sección del Proyecto HOGAR. Por último, agradezco también a la empresa *Bit&Brain Technologies, S.L.*, por haberme dado la oportunidad de formar parte de un proyecto tan interesante y novedoso, así como por haberme permitido obtener todos los conocimientos que he adquirido a lo largo de este proyecto.

Glosario y acrónimos

Glosario

Algoritmo de Welch: Técnica utilizada para estimar la densidad espectral de potencia (PSD) de una señal en el dominio de la frecuencia. Es una mejora del método clásico de estimación de la PSD mediante la Transformada Rápida de Fourier (FFT), diseñada para reducir la varianza de la estimación a costa de una resolución en frecuencia ligeramente más baja.

Amplificador: Dispositivo electrónico que aumenta la amplitud de una señal eléctrica, manteniendo su forma original.

ANOVA (ANalysis Of VAriance): Método estadístico que se utiliza para comparar las medias de tres o más grupos y determinar si existen diferencias significativas entre ellas.

Artefacto: Cualquier señal o interferencia no cerebral que se registra en los datos y que puede distorsionar o enmascarar la actividad eléctrica real del cerebro. Suelen ser provocados por factores externos, como movimientos del cuerpo, movimientos oculares como parpadeos, actividad muscular, problemas técnicos en el dispositivo EEG o incluso interferencias electromagnéticas.

AUROC: Área bajo la curva ROC, que resume en un solo valor la relación entre la tasa de verdaderos positivos y de falsos positivos para diferentes umbrales de decisión de un modelo.

Biomarcador: Característica biológica medible que indica procesos normales, patológicos o respuestas a una intervención, como un tratamiento.

Callback: Función o conjunto de funciones que se ejecutan automáticamente en puntos específicos durante el entrenamiento de un modelo. Permiten monitorizar el comportamiento del proceso de entrenamiento sin necesidad de intervenir manualmente.

Canal EEG: Conexión específica utilizada para registrar la actividad eléctrica del cerebro. Cada canal corresponde a la diferencia de potencial eléctrico registrada entre dos electrodos colocados en puntos específicos del cuero cabelludo, según un sistema de posicionamiento estandarizado. Un electrodo, el activo, registra la señal de interés, mientras que el otro, el de referencia, sirve como punto de comparación para calcular la diferencia de potencial. Cada canal produce una señal en el dominio del tiempo que refleja la actividad eléctrica generada por las neuronas del área cerebral donde esté situado el correspondiente electrodo.

Configuración de etiquetado: Criterio utilizado para asignar etiquetas o categorías a los datos en un proceso de análisis o clasificación.

Conjunto de entrenamiento: Subconjunto del conjunto de datos utilizada para entrenar un modelo.

Conjunto de validación: Subconjunto del conjunto de datos que se utiliza durante el entrenamiento de un modelo para ajustar los hiperparámetros de dicho modelo y evaluar su rendimiento de manera continua.

Conjunto de test: Subconjunto del conjunto de datos utilizado para evaluar el rendimiento final de un modelo después de que este ha sido entrenado y ajustado.

Curva de aprendizaje: Representación gráfica que muestra cómo mejora el rendimiento de un modelo de ML/DL a medida que se entrena con más datos o durante más iteraciones del proceso de optimización.

Dataset: Colección organizada de datos que se utilizan para realizar análisis, entrenar modelos de ML/DL, o llevar a cabo experimentos científicos.

Deterioro cognitivo leve (MCI - Mild Cogitive Impariment): Etapa intermedia entre el deterioro previsto de la memoria y el pensamiento, que sucede con la edad, y el deterioro más grave de la demencia. Puede incluir problemas de memoria, de lenguaje o de capacidad de juicio. Las personas que sufren de MCI pueden ser conscientes de que su memoria o sus funciones mentales han decaído.

Dispositivo EEG: Equipo médico o de investigación diseñado para medir un EEG. Cuenta con electrodos (correspondientes a los distintos canales),, amplificadores (para las señales más debiles), conversores analógico-digitales, una unidad de procesamiento para procesar las señales y, a veces, eliminar artefactos, software especializado y una fuente de alimentación.

EEG (Electroencefalograma): Prueba que mide la actividad eléctrica del cerebro. Esta prueba requiere la colocación en el cuero cabelludo de electrodos, que son unos pequeños discos metálicos. Las neuronas cerebrales se comunican mediante impulsos eléctricos, y esta actividad se manifiesta como líneas onduladas en un registro electroencefalográfico. Las neuronas cerebrales están activas todo el tiempo, incluso durante el sueño. Es uno de los estudios principales que ayudan a diagnosticar afecciones cerebrales como deterioros cognitivos, epilepsias o tumores cerebrales, entre muchas otras.

Época: Iteración completa sobre el conjunto de datos de entrenamiento durante el proceso de aprendizaje. Esto significa que, en una época, el modelo procesa todos los ejemplos de entrenamiento una vez.

Estado cognitivo: Nivel de funcionamiento de las capacidades mentales de una persona en un momento dado. Estas capacidades incluyen la memoria, la atención, el razonamiento, el juicio, el lenguaje y otras funciones relacionadas con el procesamiento de la información.

Fase del sueño: Cada una de las etapas que componen el ciclo del sueño, caracterizada por patrones específicos de actividad cerebral, movimientos oculares y tono muscular.

Feature (característica): Variable individual que representa un aspecto o propiedad de los datos y se utiliza como entrada para un modelo. Las features son la información relevante extraída de los datos que permite al modelo aprender patrones y tomar decisiones.

Fold: Partición concreta de las k particiones en las que se divide el conjunto de datos para realizar el proceso de validación.

Fototest: Test cognitivo breve que evalúa la capacidad de recordar 6 elementos que previamente se le han mostrado al sujeto y se le ha pedido que nombre. Entre denominación y recuerdo se inserta una tarea de fluidez verbal en la que el sujeto debe evocar nombres de personas agrupadas por sexo. Está especialmente indicado para la detección de sujetos con deterioro cognitivo y demencia.

Frecuencia de sampleo: Cantidad de muestras de una señal analógica que se toman por unidad de tiempo para convertirla en señal digital. Se mide en Hertzios (Hz), lo que indica el número de muestras por segundo. Por ejemplo, si la frecuencia de muestreo es 128 Hz, significa que se registran 128 muestras por segundo.

Función de pérdida: Medida matemática utilizada en el entrenamiento de modelos de ML y DL para evaluar qué tan bien se están ajustando los parámetros del modelo a los datos de entrenamiento. Cuantifica la diferencia entre las predicciones del modelo y los valores reales o esperados.

Grid Search: Enfoque sistemático para encontrar la combinación óptima de hiperparámetros para un modelo de ML/DL. Es una técnica de búsqueda exhaustiva, donde se prueba cada combinación posible de un conjunto predefinido de valores para los hiperparámetros del modelo con el objetivo de maximizar el rendimiento (medido mediante una métrica, como precisión, F1-score, etc.) en los datos de validación.

Hiperparámetro: Valor configurado manualmente antes del entrenamiento de un modelo, que no se aprende automáticamente a partir de los datos. Los hiperparámetros controlan el comportamiento del proceso de aprendizaje y la estructura del modelo. En una red neuronal, por ejemplo, puede ser el número de capas y neuronas, la tasa de aprendizaje o el tamaño del batch, entre otros.

Hipnograma: Representación gráfica del patrón de sueño de una persona a lo largo de una noche. Muestra las diferentes fases del sueño a lo largo del tiempo, en intervalos de 30 segundos, permitiendo visualizar las transiciones entre estas etapas y la duración de cada una.

kNN para imputación de datos (*k-Nearest Neighbors*): Algoritmo que rellena valores faltantes encontrando los k vecinos más cercanos (basados en una métrica de distancia) en el espacio de features y utilizando la media, mediana o moda de esos vecinos para imputar el valor faltante.

Matriz de confusión: Herramienta de evaluación utilizada en aprendizaje automático para medir el rendimiento de un modelo de clasificación. Permite visualizar las predicciones del modelo comparadas con los valores reales.

Macroarquitectura/arquitectura del sueño: Organización general de las fases y ciclos del sueño durante una noche de descanso. Esta arquitectura es un patrón regular y cíclico que se repite varias veces a lo largo de la noche, con variaciones en la duración de las diferentes fases.

Microestructura del sueño: Patrones más detallados dentro de las fases del sueño, en particular los cambios que ocurren a nivel de actividad cerebral y fisiológica a lo largo de cada fase del sueño, observables en el EEG.

Mini-Cog: Test cognitivo de memoria que consta de 3 ítems y un test de dibujo del reloj de sencilla puntuación. Puede ser utilizada de manera efectiva tras un entrenamiento breve, y su tiempo de aplicación es de unos 3 minutos.

Modelo: En el contexto de la ciencia de datos, representación matemática o computacional que intenta describir, predecir o explicar un fenómeno o patrón en los datos.

***NumPy Array*:** Estructura de datos proporcionada por la librería NumPy de Python, diseñada para almacenar y manipular grandes cantidades de datos numéricos de manera eficiente.

Ondas lentas: Tipo de actividad cerebral característica del sueño profundo con una frecuencia muy baja (0.5-2 Hz) y una amplitud alta, fundamentales para la restauración física y mental, el mantenimiento de la plasticidad cerebral, y la consolidación de la memoria.

Paper: Documento académico que presenta los resultados de una investigación original o una revisión de un tema específico en un campo del conocimiento.

Peso: Valor numérico que asigna importancia o influencia a una característica (o entrada) en el modelo, determinando su impacto en la predicción final del modelo.

Pipeline: Conjunto estructurado de pasos o procesos que se ejecutan de manera secuencial o en paralelo para completar una tarea específica. En el contexto de la ciencia de datos, la programación o la ingeniería, un pipeline se utiliza para organizar y automatizar flujos de trabajo.

Polisomnógrafo: Dispositivo médico utilizado para registrar y analizar múltiples variables fisiológicas durante el sueño de una persona. Es el equipo principal empleado en estudios de polisomnografía, una técnica de diagnóstico para evaluar trastornos del sueño.

Quejas subjetivas de memoria (SCI - Subjective Cognitive Impairment): Etapa en la que el paciente percibe problemas para recordar o retener información, de forma que afectan el desenvolvimiento cotidiano de las personas que las manifiestan. Sin embargo, y por lo general, estas personas no entran todavía en la etapa de MCI al ser clasificadas mediante diversos tests cognitivos.

Sesión: Grabación del EEG de un paciente determinado durante el sueño. Puede tener duración variable. Según el contexto que se esté tratando, puede consistir de una secuencia de fases del sueño o de una secuencia de valores correspondientes a las señales EEG.

***Spindles*:** Patrones característicos de actividad eléctrica que ocurren durante el sueño NREM, específicamente en la etapa N2. Se observan ráfagas"de ondas oscilatorias en un rango de frecuencia determinado que duran entre 0.5 y 2 segundos. Estas ráfagas"desempeñan un papel crucial en el proceso de consolidación de la memoria y la regulación del sueño.

Split: Proceso de dividir un dataset en diferentes subconjuntos con propósitos específicos durante el entrenamiento y evaluación de un modelo. En este proyecto se hablará de subconjuntos de entrenamiento, validación y test.

Test cognitivo: Herramienta diseñada para medir diferentes aspectos de las capacidades mentales o cognitivas de una persona, como la memoria, la atención, el razonamiento, la velocidad de procesamiento y las funciones ejecutivas.

Test estadístico: Procedimiento que utiliza datos de una muestra para evaluar una hipótesis sobre una población. Su objetivo principal es determinar si hay suficiente evidencia estadística para aceptar o rechazar una hipótesis planteada.

Transformada Rápida de Fourier (FFT - Fast Fourier Transform): Algoritmo eficiente que calcula la Transformada Discreta de Fourier (DFT - Discrete Fourier Transform) de una señal y su inversa. La DFT es una técnica matemática que descompone una señal en sus componentes de frecuencia, permitiendo analizarla en el dominio de la frecuencia en lugar del dominio del tiempo.

Validación cruzada (Cross Validation): Técnica utilizada en ML/DL para evaluar el rendimiento de un modelo y garantizar que sea capaz de generalizar correctamente a datos no vistos. Consiste en dividir el conjunto de datos en múltiples particiones o subconjuntos y entrenar y validar el modelo de manera iterativa, asegurando que cada dato se use tanto para el entrenamiento como para la validación, pero nunca al mismo tiempo.

Ventana de tiempo: Segmento de tiempo predefinido para analizar señales EEG, generalmente 30 segundos cuando se analizan estos datos durante la noche.

Violin plot: Representación gráfica que muestra la distribución de un conjunto de datos, destacando su forma, densidad y resumen estadístico en una sola representación

Acrónimos

α: alfa

AASM: American Academy of Sleep Medicine (Academia Americana de la Medicina del Sueño)

API: Application Programming Interface (Interfaz de Programación de Aplicaciones)

ANOVA: Analysis of Variance (Análisis de la varianza)

AUROC: Area Under Receiver Operating Characteristic Curve (Área Bajo la Curva Característica Operativa del Receptor)

AUPCR: Area Under Precision-Recall Curve (Área debajo de la Curva de Precision-Recall)

β1, β2 y β3: beta 1, beta 2 y beta 3

CPU: Central Processing Unit (Unidad Central de Procesamiento)

CV: Cross Validation (Validación Cruzada)

δ: delta

DEM: Dementia (Demencia)

DNN: Deep Neural Network (Red Neuronal Profunda)

DL: Deep Learning (Aprendizaje profundo)

DUTS: Discrete Univariable Time Series (Serie Temporal Discreta Univariable)

EEG: Electroencefalograma

FFT: Fast Fourier Transform (Transformada Rápida de Fourier)

FN: False Negatives (Falsos negativos)

FP: False positives (Falsos positivos)

γ: gamma

GPU: Graphical Processing Unit (Unidad de Procesamiento Gráfico)

GRU: Gated Recurrent Unit

GS: Grid Search (Búsqueda en malla)

H: Healthy (Sano)

HC: Hjorth Complexity (Complejidad de Hjorth)

Hz: Hertzios

IAF: Individual Alpha Frequency (Frecuencia individual en α)

LR: Logistica Regression (Regresión Logística)

LSTM: Long-Short Term Memory

MCI: Mild Cognitive Impairment (Deterioro Cognitivo Leve)

ML: Machine Learning (Aprendizaje automático)

MLP: Multi-Layer Perceptron (Perceptrón Multicapa)

OvO: One versus One (Uno contra Uno)

PSG: Polisomnógrafo

PSD: Power Spectral Density (Densidad de Potencia Espectral)

PTP: Peak-to-Peak (tiempo de pico a pico)

RBF: Radial Basis Function

RF: Random Forest

RFC: Random Forest Classifier

RP: Relative Power (Potencia Relativa)

RNN: Recurrent Neural Network (Red Neuronal Recurrente)

SCI: Subjective Cognitive Impairment (Quejas Subjetivas de Memoria)

SE: Spectral Entropy (Entropía Espectral)

SVM: Support Vector Machine (Máquina de Soporte Vectorial)

TF: Transition Frequency (Frecuencia de transición)

Θ: theta

TN: True Negatives (Verdaderos Negativos)

TP: True Positives (Verdaderos Positivos)

XGBRFC: XGBoost Random Forest Classifier

Índice de figuras

Índice de tablas

A. Anexo: Cálculo de features de la microestructura y la macroestructura

Features de la macroestructura

Feature	Abreviatura	Descripción
Time in Bed	TIB	Tiempo total del hipnograma
Total Sleep Time	TST	Tiempo total del hipnograma durmiendo (i. e. no en fase W)
Sleep Efficiency	SE	Porcentaje del hipnograma durmiendo (TST / TIB * 100)
Sleep Onset Latency	SOL	Tiempo desde el inicio hasta la primera ventana que no sea W (tiempo hasta quedarse dormido)
Sleep Period Time	SPT	Tiempo desde el que se queda dormido hasta que se despierta por última vez
Wake After Sleep Onset	WASO	Tiempo pasado despierto después de dormirse por primera vez
Snooze time	ST	Tiempo despierto al final del hipnograma (después de despertarse por última vez)
%{Phase}TIB (6 features)	p{Phase}TIB	Porcentaje en {Phase} sobre TIB
%{Phase}SPT (6 features)	p{Phase}SPT	Porcentaje en {Phase} sobre SPT
%{Phase}TST (5 features: todas menos W)	p{Phase}TST	Porcentaje en {Phase} sobre TST
Sleep Stage Transitions	SST	Número total de transiciones de una fase a otra distinta
{Phase} Awakenings (2 features: N3 y REM)	{Phase}WKN	Número total de transiciones de {Phase} a W

Long Awakenings	WKNL	Número de periodos con una duración mayor o igual a 5 minutos en fase W después de dormirse por primera vez y antes de despertarse por última vez
{Phase} Sleep Stage Transitions (5 features: todas menos Art)	{Phase}SST	Número de transiciones a {Phase}
Duration {Phase} (6 features)	dur{Phase}	Máximo tiempo seguido en fase {Phase}
Max Duration {Phase} (4 features: N1, N2, N3 y REM)	max{Phase}	Máximo tiempo seguido en fase {Phase}
Sleep Quality	SQ	Porcentaje del tiempo durmiendo en N3 o REM
Sleep Maintenance Efficiency	SME	TST / SPT * 100
{Phase} Latency (3 features: N1, N2 y REM)	Lat{Phase}	Tiempo desde el inicio del hipnograma hasta el inicio de la primera ventana en {Phase}
Sleep Cycles	CCL	Número de ciclos del sueño
Total		**49**

Tabla A.1: Features derivadas de las fases del sueño. La nomenclatura {Phase} hace referencia a una determinada fase del sueño. Por ejemplo, la feature con nombre *N2 Sleep Stage Transition* (*N2SST*) corresponde con el número de transiciones a la fase N2. Aquellas features que implican duraciones (*TIB*, *Max Duration*, *WASO*...) se miden en minutos.

A.1. Features de la microestructura

Funcionamiento del algoritmo de Welch

El algoritmo de Welch [22] es una técnica para estimar la densidad espectral de potencia (PSD) de una señal. Es una mejora del método de periodograma, diseñado para reducir la varianza de la estimación al dividir la señal en segmentos superpuestos y promediar los periodogramas de cada segmento. Consta de 4 pasos principales:

1. **Segmentación**: Dividir la señal en segmentos de igual longitud, permitiendo un traslapo (normalmente del 50 %).
2. **Ventaneo**: Multiplicar cada segmento por una ventana (por ejemplo, Hanning o Hamming) para reducir los efectos de discontinuidad en los bordes.
3. **Transformada de Fourier**: Calcular la transformada de Fourier de cada segmento ventaneado.
4. **Periodograma**: Obtener el espectro de potencia para cada segmento mediante el cálculo del periodograma.

5. **Promediado**: Promediar los periodogramas de todos los segmentos para obtener la densidad espectral de potencia (PSD) final, reduciendo la varianza de la estimación.

Dominio del tiempo

Line length
La *line length* de una señal se calcula como la suma de las diferencias absolutas entre valores consecutivos de la señal. Este cálculo sigue la siguiente fórmula:

$$\textit{Line length} = \sum_{i=1}^{N-1} |x_i - x_{i+1}|$$

donde:

- x_i es el valor de la señal en el tiempo i,
- N es el número total de puntos en la señal,
- $|x_i - x_{i+1}|$ es la diferencia absoluta entre dos puntos consecutivos de la señal.

Kurtosis
El cálculo de la kurtosis de una ventana completa de 30 segundos sigue la fórmula descrita más adelante en este mismo apartado, solo que teniendo en cuenta los valores de la señal independientemente de la banda de frecuencia, a diferencia de la kurtosis para una banda de frecuencia concreta.

Dominio de la frecuencia

Potencias relativas
La potencia relativa (RP) de una banda se define como:

$$RP = \frac{P_{\text{banda}}}{P_{\text{total}}},$$

donde:

- $P_{\text{banda}} = \sum_{f \in [f_0, f_i]} P$: es la potencia en la banda de frecuencias de interés, cuyos umbrales están definidos por f_0 como el límite inferior, y f_i como el límite superior de la banda. Por ejemplo, para la banda α, f_0 sería 8.1 Hz y f_i sería 13 Hz.
- $P_{\text{total}} = \sum_{\forall f} P$: es la potencia total en todas las frecuencias.

Por tanto, dado un número N de subventanas (en nuestro caso, N = 15, pues dividimos ventanas de 30 segundos en subventanas de 2 segundos), podemos calcular la potencia relativa mínima, máxima y media de cada banda, así como la desviación típica de dicha potencia relativa:

$$RP_{\text{min}} = \text{mín}(RP_1, RP_2, \ldots, RP_{15})$$

$$RP_{\text{max}} = \text{máx}(RP_1, RP_2, \ldots, RP_{15})$$

$$RP_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} RP_i$$

$$RP_{\text{std}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(RP_i - RP_{\text{mean}})^2},$$

donde RP_i representa la potencia relativa de la i-ésima subventana de 2 segundos.

Ratios de potencias relativas
Para calcular los ratios de las potencias relativas, se utilizan las mismas fórmulas que en el apartado inmediatamente anterior. Sin embargo, no se obtienen los valores mínimo, máximo, medio y la desviación típica sobre RP, sino sobre lo que denominaremos RP_{ratio}. Este valor se obtiene al dividir las potencias relativas de dos bandas concretas:

$$RP_{\text{ratio\_banda1\_banda2}} = \frac{RP_{\text{banda1}}}{RP_{\text{banda2}}},$$

donde RP_{banda1} y RP_{banda2} son las potencias relativas de las bandas 1 y 2, respectivamente.

Kurtosis de banda
El cálculo de la kurtosis para cada banda se ha realizado siguiendo la fórmula:

$$K = \frac{\mathbb{E}\left[(x-\mu)^4\right]}{\sigma^4}$$

donde:

- $\mathbb{E}\left[(x-\mu)^4\right]$ es el valor esperado de la cuarta potencia de las desviaciones respecto a la media.
- μ es la media de los valores de x, definida como:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- σ es la desviación estándar, definida como:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

En estas ecuaciones:

- x_i representa el valor de la i-ésima muestra de la señal.
- n es el número total de muestras.

Teniendo K, podemos calcular la kurtosis mínima, máxima, media y su desviación típica para las N (15) subventanas de cada ventana de 30 segundos:

$$K_{\text{min}} = \text{mín}(K_1, K_2, \ldots, K_{15})$$

$$K_{\text{max}} = \text{máx}(K_1, K_2, \ldots, K_{15})$$

$$K_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} K_i$$

$$K_{\text{std}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (K_i - K_{\text{mean}})^2},$$

donde K_i representa la kurtosis de la i-ésima subventana de 2 segundos.

Features individuales del EEG y potencias de sub-bandas α
Teniendo los valores de la Frecuencia de Transición (*Transition Frequency - TF*) y de la Frecuencia Individual de Alfa (*Individual Alpha Frequency - IAF*) podemos calcular los rangos de cada sub-banda de α para cada ventana de 30 segundos:

- α_1: desde TF Hz hasta (TF - IAF) / 2 Hz
- α_2: desde (TF - IAF) Hz / 2 hasta IAF Hz
- α_3: desde IAF Hz hasta IAF + 2 Hz

Ahora, para calcular la Densidad Espectral de Potencia (PSD) para cada banda, seguimos la siguiente fórmula:

$$\text{PSD}_{\text{band}} = \sum_{f_i \in [f_{\text{lower}}, f_{\text{upper}}]} P(f_i),$$

donde:

- $\text{PSD}_{\alpha\text{sub-band}}$ es la densidad espectral de potencia para la sub-banda α especificada.
- $P(f_i)$ es el valor de la densidad espectral de potencia en la frecuencia f_i de dicha sub-banda.
- El rango de frecuencias es definido por f_{lower} y f_{upper}, que representa los umbrales donde se encuentra la sub-banda α.

Cálculo de *spindles* y ondas lentas

Para obtener las distintas features relacionadas con *spindles* y ondas lentas se utilizó la librería YASA de *Python*. Sus algoritmos para la detección de *spindles* y ondas lentas, así como todas las métricas que reportan cuando detectan uno de estos eventos para una señal dada, se describen en su repositorio de *GitHub*.

B. Anexo: Modelos utilizados y experimentos totales

Descripción de los modelos utilizados

En esta sección se describirán todos los modelos utilizados a lo largo del proyecto, así como sus funciones de pérdida. Siempre que se hable en función de C (salvo en la descripción de los modelos LSTM) se hará referencia al número de clases involucradas en los experimentos con el modelo correspondiente (2 ó 4, según la configuración de etiquetado).

Modelos utilizados en los experimentos con hipnogramas como secuencias de fases del sueño

- **LSTMs**: tipo de RNN diseñada para manejar secuencias de datos y mitigar el problema del desvanecimiento del gradiente [23] que afecta a las RNN tradicionales. Su arquitectura se destaca por incluir los siguientes elementos:
  - Celdas de memoria: permite a la red "recordar" u "olvidar" información a lo largo del tiempo.
  - Puertas: pueden ser de entrada, para controlar qué nueva información se añade a la celda de memoria; de olvido, para decidir qué información debe ser descartada; o de salida, para determinar qué parte de la infroamción almacenada en la celda pasa al siguiente estado oculto.
  - Estados: pueden ser de celda (Ct), los cuales contienen la memoria a largo plazo, u ocultos (ht), que representan la memoria a corto plazo y sirve como salida de la celda en un tiempo específico.

  Con estos elementos, las LSTMs son capaces de manejar dependencias a largo plazo en datos secuenciales, como los hipnogramas. Durante el entrenamiento, las puertas permiten filtrar y actualizar la información de manera eficiente, evitando que datos irrelevantes saturen la memoria o que información importante se pierda. La Figura B.1 muestra cómo funciona una neurona LSTM internamente.

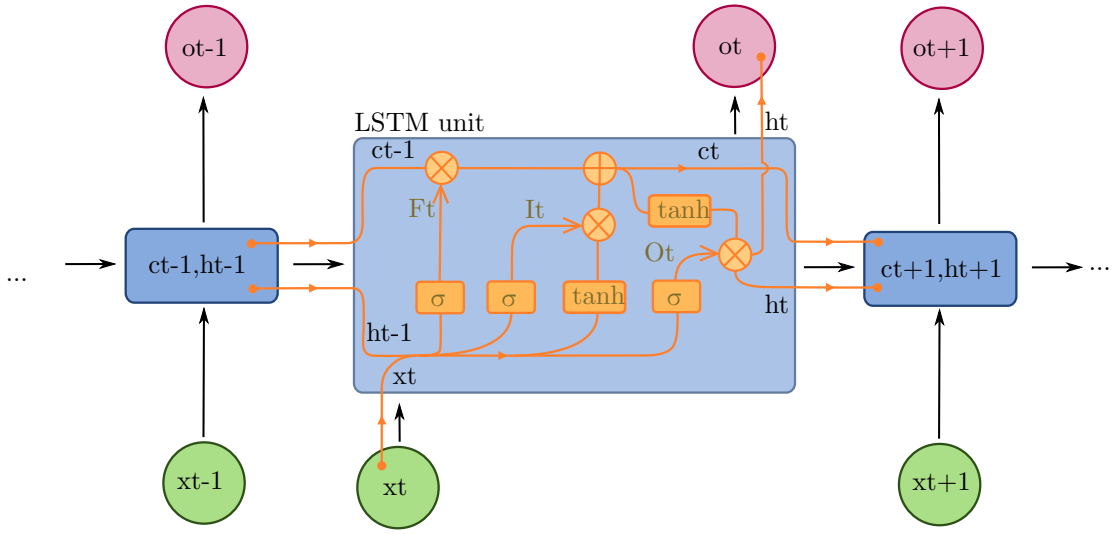


Figura B.1: Funcionamiento de una neurona LSTM (imagen de *Wikimedia Commons*)

- **GRU**: variante simplificada de las LSTMs que reducen la complejidad computacional al combinar ciertos componentes. Su arquitectura incluye los siguientes elementos:
  - **Celdas de memoria**: almacenan información relevante de la secuencia y permiten actualizar o descartar datos según sea necesario.
  - **Puertas**:
    - **Puerta de actualización**: controla cuánto de la información pasada debe ser conservada y cuánto debe ser actualizada con la nueva información.
    - **Puerta de reinicio**: decide qué parte de la memoria previa debe ser olvidada al procesar el estado actual.
  - **Estado oculto** (h_t): combina las funciones del estado de celda y el estado oculto de las LSTMs, simplificando la arquitectura y reduciendo los cálculos necesarios.

  Con estos elementos, las GRU también son capaces de capturar dependencias a largo plazo en datos secuenciales, pero con menor demanda de recursos computacionales. Durante el entrenamiento, las puertas de actualización y reinicio trabajan juntas para ajustar de manera eficiente qué información se retiene o descarta, permitiendo un aprendizaje eficaz en tareas como la predicción o clasificación de series temporales. La Figura B.2 muestra cómo funciona una neurona GRU internamente.

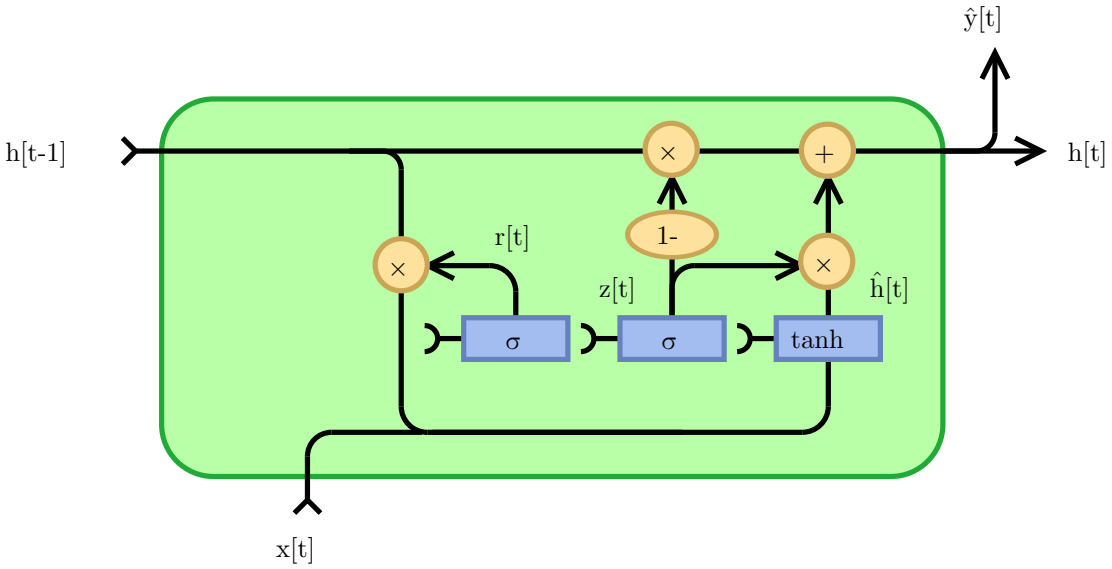


Figura B.2: Funcionamiento de una neurona GRU (imagen de *Wikimedia Commons*)

En este trabajo, se han probado 12 modelos distintos utilizando RNNs:

- LSTM1: 1 capa recurrente, 16 neuronas LSTM en esa capa
- LSTM2: 1 capa recurrente, 32 neuronas LSTM en esa capa
- LSTM3: 1 capa recurrente, 64 neuronas LSTM en esa capa
- LSTM4: 3 capas recurrentes, 16 neuronas LSTM en cada capa
- LSTM5: 3 capas recurrentes, 32 neuronas LSTM en esa capa
- LSTM6: 3 capa recurrente, 64 neuronas LSTM en esa capa
- GRU1: 1 capa recurrente, 16 neuronas GRU en esa capa
- GRU2: 1 capa recurrente, 32 neuronas GRU en esa capa
- GRU3: 1 capa recurrente, 64 neuronas GRU en esa capa
- GRU4: 3 capas recurrentes, 16 neuronas GRU en cada capa
- GRU5: 3 capas recurrentes, 32 neuronas GRU en esa capa
- GRU6: 3 capa recurrente, 64 neuronas GRU en esa capa

Todos estos modelos cuentan con una capa de *embedding* como capa de entrada. El propósito de esta capa especial es convertir valores discretos en vectores de embedding densos, de forma que la capa recurrente sea capaz de trabajar con estos valores. Además, debido a que las RNNs solo trabajan con vectores de entrada de longitud fija, se asignan "0s" a las ventanas faltantes de aquellas sesiones de menos de 9 horas. De esta forma, la dimensión de entrada de esta capa de *embedding* será 7, una por cada fase del sueño (5), una para la fase que representa artefactos y una para estos "0s" insertados.

La última capa recurrente de estos modelos RNN debe ir conectada a una capa densa, que en nuestro caso será de 32 neuronas, para posteriormente conectarla con la capa de salida.

Un ejemplo de la arquitecura de estos modelos (LSTM4) se observa en la Figura B.3. Este ejemplo podría aplicarse también a los modelos con capas GRU. Para entender la disposición de neuronas en una capa cualquiera (densa, LSTM o GRU), véase Figura B.5.

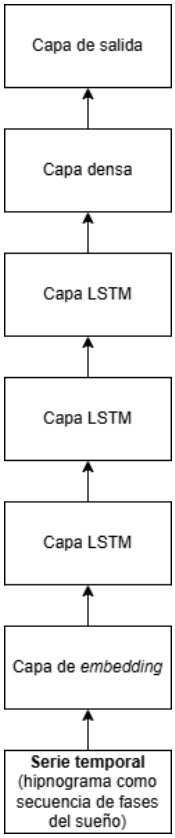


Figura B.3: Ejemplo de RNN utilizada

- **Modelo híbrido**: el modelo híbrido combina capas densas con capas LSTMs mediante capas de concatenación. De esta forma, podemos introducir información tabular (en nuestro caso, la edad y las puntuaciones en 4 tests cognitivos) junto a información dispuesta en forma de serie temporal (en nuestro caso, la evolución de las fases del sueño a lo largo del tiempo). La Figura B.4 muestra un diagrama, a alto nivel, de la arquitectura de este modelo con 2 concatenaciones.

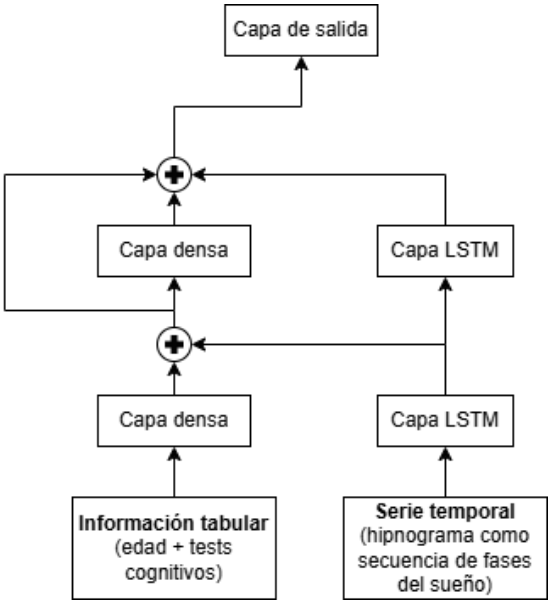


Figura B.4: Arquitectura del modelo híbrido

Al ser estos modelos diferente versiones de redes neuronales, la función de pérdida utilizada para todos ellos se describe al final de la siguiente subsección.

Modelos utilizados en los experimentos con hipnogramas como features derivadas de la microestructura y la macroestructura

Máquina de soporte Vectorial (*Support Vector Machine*, SVM)

Este modelo busca encontrar un hiperplano óptimo que separe los datos en diferentes clases. Su funcionamiento se basa en los siguientes componentes clave:

- **Hiperplano**: es la frontera de decisión que separa las clases de manera óptima en el espacio de características. En problemas linealmente separables, se busca maximizar el margen, es decir, la distancia entre el hiperplano y los puntos más cercanos de cada clase.
- **Vectores de soporte**: son los puntos de datos más cercanos al hiperplano. Estos determinan su posición y orientación, siendo esenciales para la definición del modelo.

  Además, a lo largo de este proyecto se ha experimentado con diferentes valores para los siguientes parámetros:

  - **Funciones kernel**: permiten al SVM trabajar con datos no linealmente separables al proyectarlos a un espacio de mayor dimensionalidad. Los kernels más comunes son el lineal, para datos linealmente separables; el polinómico, encontrar relaciones más complejas entre las características; y el *RBF* (*Radial Basis Function*), para manejar datos con límites de decisión complejos.

- C: controla el equilibrio entre maximizar el margen y minimizar los errores de clasificación. Valores grandes de C buscan clasificar correctamente todos los puntos, mientras que valores pequeños permiten más errores a favor de un margen más amplio.
- γ: en kernels no lineales (como RBF), define la influencia de un único punto de datos. Valores altos de γ generan fronteras más complejas, mientras que valores bajos las hacen más suaves.

- **Función de pérdida**: la función de pérdida del modelo SVM utilizada durante este proyecto ha sido la función *Hinge Loss*. Concretamente, ha sido la implementación para la clase *SVC* de *Scikit-Learn*. Esta se puede definir para los problemas de clasificación binaria (y para el conjunto de todas las sesiones) en función de w_i (vector de probabilidades de las decisiones del modelo, de tamaño C) como:

$$L_{Hinge}(y, w) = \frac{\sum_{i=0}^{N-1} \text{máx}\,(0, 1 - w_i \cdot y_i)}{N}$$

  Para los problemas de clasificación multiclase, la *Hinge Loss* utiliza la variante de Crammer & Singer [24]:

$$L_{Hinge}(y, w) = \frac{\sum_{i=0}^{N-1} \text{máx}(0, 1 + \hat{w}_{i,y_i} - w_{i,y_i})}{N}$$

El SVM funciona asignando un conjunto de pesos a las características y ajustando el hiperplano iterativamente durante el entrenamiento. Este enfoque es útil en problemas como la clasificación de señales EEG o de hipnogramas caracterizadas como features tabulares, ya que puede encontrar fronteras no lineales en datos complejos y de alta dimensionalidad, asegurando una buena generalización.

Regresión Logística (*Logistic Regression*, LR)

Modelo principalmente utilizado para tareas de clasificación binaria, aunque puede extenderse a múltiples clases (regresión logística multinomial). Se basa en los siguientes componentes y parámetros:

- **Función lineal**: combina las características de entrada ($X = [x_1, x_2, ..., x_n]$) con pesos ($W = [w_1, w_2, ..., w_n]$) y un sesgo (b) para formar una suma ponderada:

$$z = W \cdot X + b$$

- **Función sigmoide**: transforma la salida z en una probabilidad entre 0 y 1:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

  Esta probabilidad se utiliza para clasificar las muestras en una de las dos clases (usualmente 0 o 1), utilizando un umbral predeterminado (generalmente 0.5).

- **Optimización**: utiliza algoritmos como el descenso del gradiente para ajustar los pesos (W) y el sesgo (b), minimizando la pérdida logarítmica.

- **Regularización**: ayuda a prevenir el sobreajuste mediante términos adicionales que penalizan pesos grandes.
- **Función de pérdida**: la función de pérdida del modelo LR utilizada durante este proyecto ha sido la función *Log Loss*, más conocida como *Cross-Entropy Loss* (Entropía Cruzada). Se ha utilizado la versión implementada para la clase *LogisticRegression* de *Scikit-Learn*. Para el conjunto de todas las sesiones, en problemas de clasificación tanto binaria como multiclase su fórmula es:

$$L_{Cross\text{-}Entropy}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

El funcionamiento de la regresión logística consiste en calcular la probabilidad de cada clase para cada muestra, compararla con un umbral y asignar la clase correspondiente. Además, con estrategias como la regularización se trata de prevenir el sobreajuste mediante términos adicionales que penalizan pesos grandes. En este proyecto, la regularización utilizada ha sido la *L2* [25], que distribuye los pesos uniformemente para evitar que crezcan demasiado. Debido a su simplicidad, capacidad de interpretación y eficacia en conjuntos de datos linealmente separables, este modelo es ampliamente utilizado en problemas de clasificación.

Clasificador mediante bosque de árboles de decisión (*Random Forest Classifier*, RFC)

Basado en ensambles, combina múltiples árboles de decisión para realizar tareas de clasificación. Su robustez y capacidad de generalización lo hacen especialmente útil para datos complejos y ruidosos. Cuenta con los siguientes componentes clave:

- **Árboles de decisión**: cada árbol se entrena en una muestra aleatoria del conjunto de datos y crea particiones sucesivas basadas en las características que mejor separan las clases. Las decisiones se toman evaluando divisiones en las características hasta llegar a una hoja que asigna una clase.
- ***Bootstrap* (muestras con reemplazo)**: cada árbol se entrena con un subconjunto aleatorio de las muestras del conjunto de entrenamiento, seleccionadas con reemplazo. Esto introduce diversidad en el *ensamble*.
- **Selección aleatoria de características**: en cada división del árbol, solo se considera un subconjunto aleatorio de las características, lo que reduce la correlación entre árboles y mejora la generalización.
- **Votación por mayoría**: para tareas de clasificación, cada árbol predice una clase, y la clase final se determina por el voto mayoritario de todos los árboles.

  Los valores utilizados de los demás hiperparámetros, como el número de árboles, la profundidad máxima o el número de características han sido aquellos que se encuentran por defecto en los modelos ya desarrollados importados a través de las librerías mencionadas.
- **Criterio de decisión (función de pérdida)**: los árboles de decisión trabajan con funciones para medir la "calidad" de un split, denominados criterios de decisión. Para ello, debemos describir la proporción (p_{mk}, en función de la clase k predicha entre las C clases)

de sesiones de una clase determinada "observada" por un nodo m. Una vez contamos con esta proporción y con los datos en el nodo m (Q_{mk}), podemos definir 2 criterios diferentes:

- **Impureza de Gini**:
$$L_{Gini}(Q_m) = \sum_k p_{mk} \cdot (1 - p_{mk})$$

- **Entropía (*Log Loss*)**:
$$L_{Entropy}(Q_m) = -\sum_k p_{mk} \cdot \log(p_{mk})$$

Clasificador mediante bosque de árboles de decisión basado en el algoritmo XGBoost (*Random Forest Classifier based on XGBoost algorithm*, XGBRFC)

Similar al clasificador que utiliza *Random Forest* que se acaba de describir, este modelo hace uso de la implementación del algoritmo XGBoost siendo ideal para tareas de clasificación en datos complejos. a diferencia de un *Random Forest Classifier* tradicional, XGBRFC utiliza un esquema ponderado para combinar las predicciones de los árboles, aprovechando la implementación eficiente de *XGBoost*. Incluye características como la gestión eficiente de memoria, regularización explícita y paralelización, que hacen al modelo más rápido y escalable en comparación con un *Random Forest Classifier* estándar. En este proyecto, se han utilizado los mismos criterios de decisión que con el modelo RFC.

Red Neuronal Profunda (*Deep Neural Network*, DNN)

Modelo compuesto por múltiples capas de neuronas conectadas entre sí, diseñado para aprender patrones complejos en los datos mediante representaciones jerárquicas. Sus principales componentes son:

- **Capas de entrada**: recibe las características del conjunto de datos ($X = [x_1, x_2, ..., x_n]$) y las pasa a través de las conexiones con la primera capa oculta.
- **Secuencia de capas ocultas**: consiste en una o más capas de neuronas que transforman las entradas mediante operaciones lineales seguidas de funciones de activación no lineales. Estas capas extraen características jerárquicas y complejas de los datos:
$$h^{(l)} = \sigma(W^{(l)} \cdot h^{(l-1)} + b^{(l)})$$
Donde $h^{(l)}$ es la salida de la capa l, $W^{(l)}$ son los pesos, $b^{(l)}$ es el sesgo y σ es la función de activación.
- **Capa de salida**: genera las predicciones finales. La activación utilizada depende de la tarea:
  - **Clasificación binaria**: función sigmoide para generar probabilidades entre 0 y 1.
  - **Clasificación multiclase**: función softmax para probabilidades de múltiples clases. Es la función que se ha utilizado en todos los experimentos del proyecto. Dado un vector de entrada $z = [z_1, z_2, \ldots, z_n]$, esta función de activación se define como:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad \text{para} \quad i = 1, 2, \ldots, n$$

donde:

- z_i es el valor de la entrada para la i-ésima clase.
- e^{z_i} es la exponencial de z_i.
- La suma en el denominador es la normalización para asegurar que la salida de la función Softmax sea una distribución de probabilidad, es decir, que la suma de todas las salidas sea igual a 1.

La salida de la función *softmax* es un vector de probabilidades, donde cada elemento del vector es una probabilidad que indica la pertenencia a una clase específica.

En el caso de realizar una clasificación binaria, la última capa tendrá una sola neurona si la función de activación es la función sigmoide, y 2 neuronas si la función es softmax. Si la clasifiación es multiclase, la última capa utilizará la función de activación softmax con 4 neuronas.

- **Función de pérdida**: mide el error entre las predicciones y los valores reales, orientando el ajuste de los pesos.
- **Regularización**: técnicas como *dropout* o penalizaciones *L1/L2* [25] ayudan a prevenir el sobreajuste al reducir la complejidad del modelo.
- **Hiperparámetros importantes**:
  - **Número de capas ocultas**: define la profundidad de la red.
  - **Número de neuronas por capa**: controla la capacidad de aprendizaje de cada capa.
  - **Función de activación**: determina la no linealidad; ejemplos comunes incluyen *ReLU*, sigmoide y *tanh*.
  - **Tasa de aprendizaje**: regula la velocidad de actualización de los pesos.
  - ***Dropout***: fracción de neuronas desconectadas aleatoriamente durante el entrenamiento para mejorar la generalización.
  - **Algoritmo de optimización**: utiliza algoritmos como el descenso estocástico del gradiente (*SGD*) o sus variantes (*Adam, RMSProp*) [26] para ajustar los pesos y el sesgo, minimizando la función de pérdida.

Una red neuronal profunda simple funciona procesando las entradas a través de múltiples capas de neuronas interconectadas, aprendiendo representaciones jerárquicas y complejas de los datos. Es especialmente útil para problemas como la clasificación de señales EEG o hipnogramas, donde los patrones son altamente no lineales y multidimensionales.

En este proyecto, se han utilizado dos arquitecturas diferentes de DNNs. La arquitectura DNN1 cuenta con 4 capas ocultas, con 32, 32, 16 y 16 neuronas respectivamente (Figura B.5). La arquitectura DNN2 cuenta con 3 capas, con 64, 64 y 32 neuronas respectivamente.

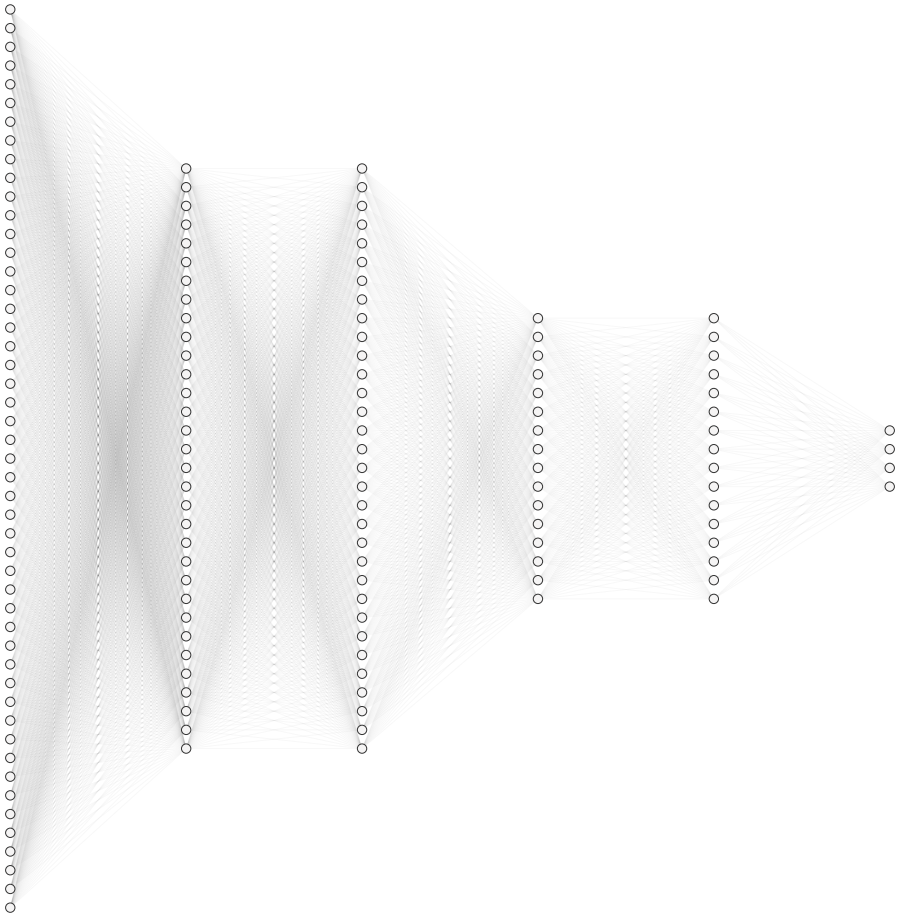


Figura B.5: Ejemplo de arquitectura del modelo DNN1. En la figura, se utilizan 49 neuronas en la capa de entrada y 4 en la de salida. Por tanto, esta arquitectura en concreto ha sido utilizada para tratar los las features derivadas de los hipnogramas, para clasificar las 4 clases. En este diagrama no están representada la normalización de los pesos (capas *BatchNormalization*) ni la estrategia *Dropout*, ya que estas funciones se aplican directamente sobre las capas ocultas [27]. La herramienta utilizada para crear este diagrama ha sido *NN-SVG*.

Funciones de pérdida de los modelos con redes neuronales

En los modelos que utilizan redes neuronales (tanto RNNs como DNNs), la función de pérdida utilizada ha sido la Entropía Cruzada Focal. Concretamente, la implementación utilizada ha sido una versión basada en un parámetro α , desarrollada por Scikit-Learn. Esta se puede describir mediante la siguiente expresión para un solo sujeto:

$$L_{FocalCrossEntropyAlpha\text{-}Balanced}(y, \hat{y}) = -\alpha \cdot (1 - p_t)^{\gamma} \cdot \text{CrossEntropy}(y, \hat{y})$$

En esta expresión, α es el factor de ponderación de los pesos de las clases, teniendo todas las

clases los mismos pesos si su valor fuera 1. El parámetro de enfoque γ reduce la importancia dada a ejemplos de la clase mayoritaria (en nuestro caso, sesiones de personas sanas) que sean fáciles de clasificar. La función (p_t) se puede definir como:

$$p_t = \begin{cases} \text{output}, & \text{si } y = 1, \\ 1 - \text{output}, & \text{en otro caso.} \end{cases}$$

Los valores utilizados tanto de α como de γ han sido aquellos que venían por defecto en la librería de *Keras* utilizada ($\alpha = 0.25$ y $\gamma = 2.0$). La función original de la Entropía Cruzada Focal sigue la fórmula:

$$L_{FocalCrossEntropy}(y, \hat{y}) = -\sum_{i=1}^{n} (1 - p_i)^{\gamma} \log_b(p_i) \tag{B.1}$$

Por otro lado, la tasa de aprendizaje sigue una estrategia, denominada decaimiento por pasos, de forma que su valor se va ajustando de manera dinámica a lo largo del entrenamiento de los modelos de DL de este trabajo. El valor de la tasa de aprendizaje *lr* en un momento t según esta estrategia de decaimiento por pasos viene dado por la siguiente fórmula:

$$\text{lr}(t) = \text{initial\_learning\_rate} \cdot \text{decay\_rate}^{\left\lfloor \frac{\text{step}}{\text{decay\_steps}} \right\rfloor} \tag{B.2}$$

donde:

- *initial_learning_rate*: valor inicial de la tasa de aprendizaje (p. ej., 0.0001)
- *decay_rate*: factor de decaimiento. Su valor para todos los experimentos ha sido de 0.96.
- *step*: paso (época de entrenamiento) en el momento t.
- *decay_steps*: número de pasos totales de decaimiento.

Cálculo de experimentos totales realizados

En esta sección se detalla el número total de experimentos realizados en base a los modelos utilizados en cada forma de modelar el problema, que denominaremos *approach*. La Tabla B.1, que desglosa estos cálculos, contabiliza los experimentos realizados con modelos de ML (SVM, LR, RFC y XGBRFC) por un lado, y por otro los realizados con modelos de DL (redes LSTM, redes GRU y DNNs, dependiendo del *approach* correspondiente). Es importante remarcar que este desglose no tiene en cuenta los conjuntos de hiperparámetros utilizados en los bucles de validación de los modelos de DL, es decir, aquellos que prueban distintos valores para las tasas de aprendizaje, *batch sizes* o incluso los distintos algoritmos de optimización.

Los experimentos del *approach* 1 se han realizado con 2 arquitecturas de RNNs diferentes, LSTMs y GRUs. Estas 2 arquitecturas contaban con 3 números distintos de neuronas en sus capas recurrentes, y con 2 números de capas distintos. Además, se realizaron también experimentos con el modelo híbrido, que utilizó una serie de concatenaciones de capas LSTM y densas. Estos modelos contaron con 3 valores distintos para el número de neuronas en sus capas LSTM y densas, así como 3 números de concatenaciones distintos. Los experimentos en este *approach* no

han utilizado modelos de ML debido a la incapacidad de los modelos escogidos para tratar con series temporales.

Los experimentos del *approach* 2 han utilizado modelos tanto de ML como de DL. Respecto a los modelos de ML se ha experimentado con SVMs con 3 *kernels* distintos (lineal, polinómico de grado 3 y *RBF*), con un modelo de LR y con 2 índices de pureza distintos (Gini y entropía) para los modelos basados en RF (RFC y XGBRFC). Respecto a los modelos de DL, se han utilizado 2 arquitecturas distintas de DNN DNN1 y DNN2, descritas en este mismo Anexo B.

Los experimentos del *approach* 3 se han realizado con los mismos modelos del *approach* 2, tanto de ML como de DL. Sin embargo, se ha procedido de 4 formas distintas para realizar la correspondiente selección de features (columna "SF" en la Tabla B.1):

- Utilizar solo las features reportadas como importantes por el paper de referencia [3] a la hora de clasificar en 2 clases (sanos frente a demencia por un lado, y sanos frente a MCI por otro), antes de la selección de features basada en los coeficientes de importancia de feeatures de cada modelo.
- Realizar un estudio ANOVA seleccionando las features más discriminativas en base a un nivel de significancia (p-valor) de 0.05, antes de la selección de features basada en los coeficientes de importancia de features de cada modelo.
- Extraer directamente las features más importantes basándose en los coeficientes de importancia de features de cada modelo.

Approach	**Modelos DL**	**Modelos ML**	**SF**	**Expresión**	**Total**
1	$2 \cdot (3 \cdot 2) + (3 \cdot 3 \cdot 3)$	0	-	$(12 + 27) + 0$	**39**
2	$3 \cdot 3 + 1 + 1 \cdot 2 + 1 \cdot 2$	2	-	$(9 + 1 + 2 + 2) + 2$	**16**
3	$3 \cdot 3 + 1 + 1 \cdot 2 + 1 \cdot 2$	2	4	$4 \cdot ((9 + 1 + 2 + 2) + 2)$	**64**
Total (por configuración)					**119**

Tabla B.1: Desglose de todos los experimentos realizados durante el proyecto para una sola configuración de etiquetado. Los valores de la columna "Approach" son "1" para los experimentos correspondientes a tratar las sesiones como la evolución de las fases del sueño a lo largo del tiempo; "2" para los correspondientes a caracterizar las sesiones como una serie de features derivadas de la macroestructura; y "3" para caracterizarlas como una serie de features derivadas tanto de la microestructura como de la macroestructura. Por tanto, al contar con 3 configuraciones de etiquetado, se han realizado 357 experimentos.

C. Anexo: Otros resultados relevantes

Resultados de los experimentos combinando microestructura y macroestructura, utilizando ANOVA antes de la selección de features en base a los coeficientes de importancia

En las Tablas C.1 y C.2 se pueden visualizar los resultados (*test accuracies* y *balanced test accuracies*, respectivamente) procedentes de realizar los experimentos combinando las features de la microestructura y macroestructura, utilizando el estudio estadístico ANOVA ($p-valor < 0.05$) para la preselección de features.

Modelo	H vs SCI, MCI y DEM	H y SCI vs MCI y DEM	4 clases
SVM	59.72	73.01	51.19
LR	55.22	68.05	45.74
RFC	65.37	78.53	64.39
XGBRFC	61.19	78.98	62.08
DNN1	54.85	63.07	25.58
DNN2	56.62	58.42	25.13

Tabla C.1: *Test accuracies* (%) de los modelos probados combinando micro y macroestructura, utilizando ANOVA para la preselección de features.

Modelo	H vs SCI, MCI y DEM	H y SCI vs MCI y DEM	4 clases
SVM	60.42	68.90	53.71
LR	54.38	63.35	47.21
RFC	60.64	53.46	35.90
XGBRFC	56.24	58.78	32.82
DNN1	55.78	55.92	36.60
DNN2	52.45	60.49	40.32

Tabla C.2: *Balanced test accuracies* (%) de los modelos probados combinando micro y macroestructura, utilizando ANOVA para la preselección de features.

D. Anexo: Estudios paralelos

Influencia de los tests cognitivos y la edad

Debido a los malos resultados al comienzo de este proyecto utilizando modelos de RNNs, la empresa *Bitbrain* facilitó información acerca de las puntuaciones de los pacientes en distintos tests cognitivos (MoCA, Fototest, Minicog, y el índice *IR_FCSRT* [28]), llevados a cabo por profesionales de la empresa, además de la edad. Estos tests tratan de obtener información acerca de las distintas facultades mentales de los sujetos, principalmente relacionadas con los distintos tipos de memoria (semántica, episódica...), la atención o la capacidad de aprendizaje. Por tanto, tanto la puntuación de estos tests como la edad son variables que están muy estrechamente relacionadas con el deterioro cognitivo de las personas mayores. La Figura D.1 permite visualizar las distribuciones de algunas de estas variables en función de su estado cognitivo, mostrando una clara diferencia entre dichos estados.

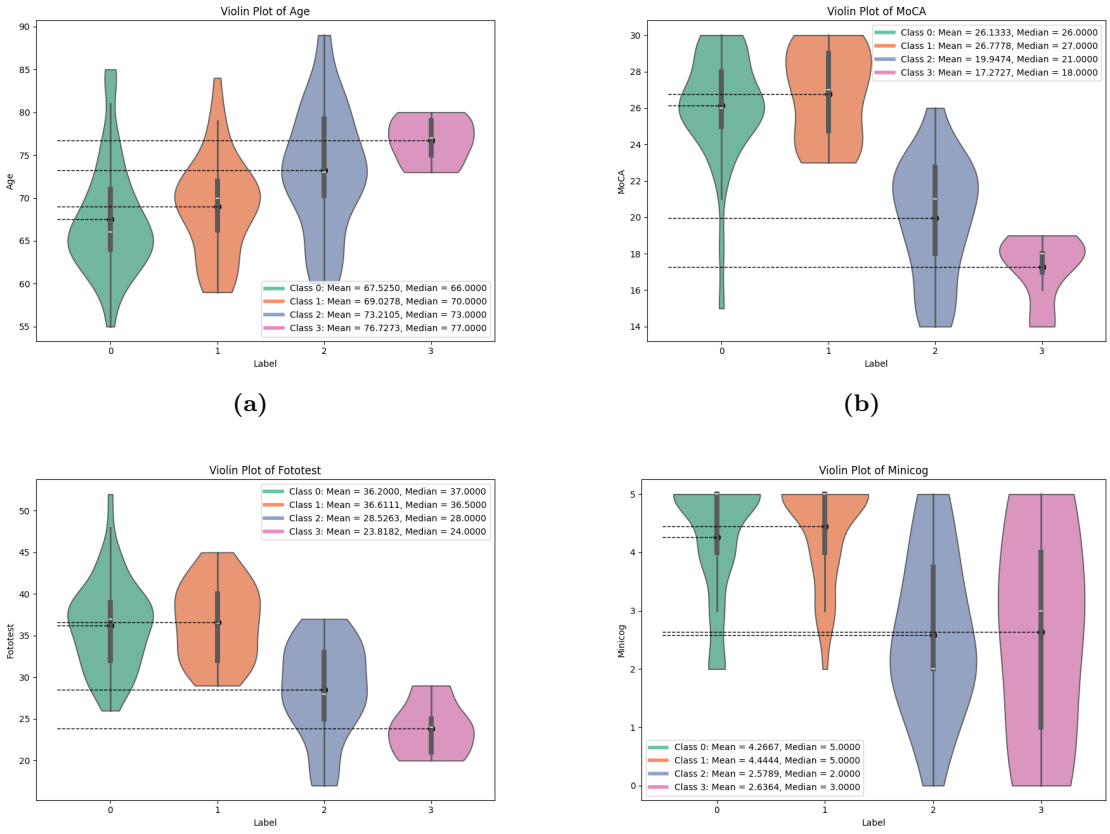


Figura D.1: Distribución de la edad (subfigura a)), puntuación en el test *MoCA* (subfigura b)), puntuación en el test Fototest (subfigura c)) y puntuación en el test Minicog (subfigura d)) de los pacientes, según su estado cognitivo. Los valores de "Label" son "0" para sujetos sanos, "1" para pacientes con SCI, "2" para pacientes con MCI y "3" para pacientes con demencia (DEM).

Por tanto, al contar con estas nuevas 5 variables, se trató de visualizar qué importancia tenía esta información a la hora de que los modelos utilizados en este proyecto aprendieran a clasificar el estado cognitivo. Un ejemplo muy claro de diferencia de resultados, utilizando la configuración de etiquetado 2, es con el uso de clasificadores Random Forest. La Figura D.2 muestra la diferencia de resultados al usar estos tests y la edad para el modelo RFC, frente a no usar esta información.

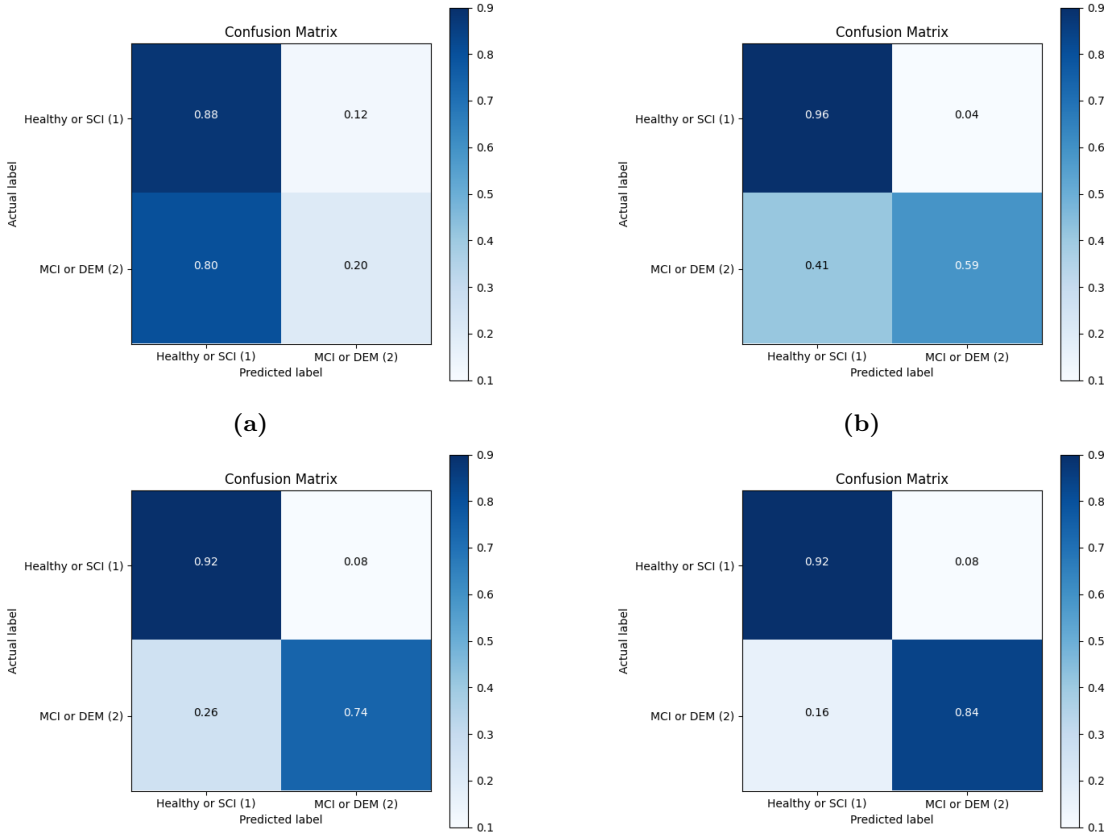


Figura D.2: Diferencia de resultados del modelo RFC, utilizando como criterio de división el índice Gini, al utilizar la edad y los tests cognitivos. La subfigura a) muestra la matriz de confusión resultante de entrenar el modelo con las 52 features derivadas de los hipnogramas. La subfigura b) muestra la *CM* resultante de entrenarlo con las 52 features del hipnograma, además de los tests cognitivos y la edad (57 features). La subfigura c) incluye solo las 10 features más importantes del hipnograma, además de los tests cognitivos y la edad (15 features). Finalmente, la subfigura d) muestra los resultados de entrenar el modelo únicamente con los tests cognitivos y la edad (5 features).

Por otro lado, la Figura D.3 muestra la gran diferencia de importancia que el modelo le da tanto a los tests cognitivos como a la edad para tomar sus decisiones de clasificación respecto a aquellas features obtenidas directamente de la macroestructura del sueño. Estas importancias se han obtenido mediante los valores de los coeficientes asignados por el modelo RFC a las distintas features, tras ser entrenado con el conjunto completo de features para la configuración 2.

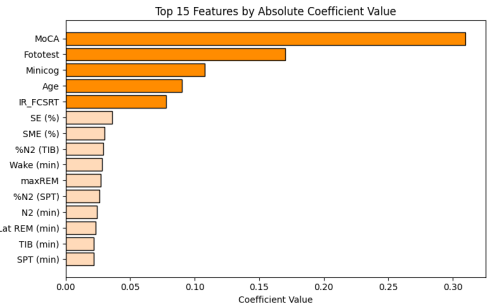


Figura D.3: Diferencia de importancia de features, según el modelo RFC (con índice Gini) entre los tests congitivos y la edad respecto a las features derivadas de los hipnogramas

Features obtenidas del EEG siguiendo siguiendo investigación que analiza la actividad cerebral durante el día

Antes de comenzar a obtener features de la microestructura del sueño, se obtuvieron una serie de features reportadas por el paper [29]. Estas features, cuyo objetivo es realizar una clasificación multiclase de personas sanas, con MCI y con AD, utiliza una serie de features obtenidas en ventanas de 4 segundos sobre grabaciones de 6 minutos durante el día. Además, utilizan 16 canales.

Ya que el dataset de este proyecto era compatible con las features reportadas por dicho paper, se obtuvieron las siguientes features a partir de los ficheros *.npz* que representan las grabaciones durante la noche. En nuestro caso, se han obtenido estas features para ventanas de 30 segundos.

Potencia relativa de banda (*Relative Power - RP*)

La potencia relativa de una banda determinada se calcula de la misma manera que en la microestructura del sueño, descrita en este Anexo A.

Entropía espectral (*Spectral Entropy - SE*)

Representa la entropía de *Shannon* del espectro de potencia de la señal completa. Esta describe la uniformidad de la distribución del espectro de potencia y, por tanto, la irregularidad del EEG. Tendremos, por ende, 2 valores (1 * 2 canales) por cada ventana de la sesión. Esta entropía se puede calcular de la siguiente manera:

$$SE = -\sum_{f} S(f) \cdot \log_2 S(f) \tag{D.1}$$

, donde S representa la potencia normalizada del espectro.

Complejidad de Hjorth (*Hjorth Complexity - HC*)
Es uno de los 3 parámetros de Hjorth (actividad, movilidad y complejidad) [30]. Se calcula como el ratio entre la movilidad de la primera derivada de la señal respecto a la movilidad de la propia señal, sin derivar. Obtendremos, por tanto, 2 valores (1 * 2 canales) por cada ventana de la sesión. Su fórmula es:

$$HC = \frac{\sigma''_s/\sigma'_s}{\sigma_s/\sigma'_s} \tag{D.2}$$

Tras calcular estas features para cada ventana de 30 segundos, contamos con 16 features por ventana, teniendo en cuenta ambos canales. De esta forma, por cada sesión construimos una matriz, cuyas dimensión viene dada por la expresión:

$$\textit{número de ventanas} \times \textit{número de features}$$

Posteriormente, siguiendo el paper de referencia en este apartado, se promediaron las filas en bloques de 12 filas para reducir la dimensionalidad del conjunto de datos, esperando que así se obtuviesen mejores resultados. En caso de que el número de ventanas no sea múltiplo de 12, las filas sobrantes se promediaban entre sí para ser añadidas como una única fila a la matriz. Se eligió este valor para promediar debido a que era el que mejor resultados reportaba en el paper (también se reportaban los valores 6, 8 y 10).

Debido a las nuevas dimensiones del dataset (3 dimensiones, en vez de 2 como habitualmente, ya que cada sesión es una matriz), el conjunto de entrenamiento con el que finalmente se entrenará el modelo se creó de la siguiente manera:

Primero, se elige aleatoriamente una sesión. Dentro de la sesión, se elige una ventana (determinada por un índice aleatorio, que será multiplicado por un parámetro *batch_size* para finalmente elegir la ventana), y se toman las siguientes *batch_size* ventanas para añadirlas al conjunto de entrenamiento final. Una vez realizado este procedimiento, se realiza de nuevo hasta recorrer de manera aleatoria todo el subconjunto de entrenamiento inicial. El conjunto de validación final se construye de la misma manera, asegurando siempre la correspondencia entre las ventanas seleccionadas y sus etiquetas de estado cognitivo.

Tras realizar por completo estos procedimientos (tanto para entrenamiento como para validación), ambos subconjuntos tendrán 3 dimensiones: *(x, y, z)*. Debido a que la red utilizada solo admite como entrada arrays de 2 dimensiones, fue necesario "aplanar" las primeras 2 dimensiones, conviertiendo los subconjuntos de 3 dimensiones (x, y, z) a 2 dimensiones (u, v).

Los resultados de los experimentos utilizando estas manera de estructurar el dataset no fueron buenos, por lo que no se reportan en este trabajo. La principal hipótesis para explicar estos resultados es que las features utilizadas provienen de un paper centrado en el análisis de EEG durante el día. Además, los valores de estas features fueron promediados en conjuntos de 12 ventanas de 30 segundos consecutivas, sin tener en cuenta las fases del sueño de dichas ventanas para agruparlas.