



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica y Automática Industrial

TRABAJO FIN DE GRADO

**MÉTODOS PARAMÉTRICOS Y REDES
NEURONALES RECURRENTE EN EL
DIAGNÓSTICO Y PRONÓSTICO DE LA
FASE PRECLÍNICA DE LA ENFERMEDAD
DE ALZHEIMER**

Jorge Bengoa Pinedo

Tutor: Carlos Platero Dueñas

Departamento: Ingeniería Eléctrica, Electrónica, Automática y Física
Aplicada

Madrid, Junio, 2025



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica y Automática Industrial

TRABAJO FIN DE GRADO

MÉTODOS PARAMÉTRICOS Y REDES
NEURONALES RECURRENTE EN EL
DIAGNÓSTICO Y PRONÓSTICO DE LA
FASE PRECLÍNICA DE LA ENFERMEDAD
DE ALZHEIMER

Firma Autor

Firma Tutor

Copyright ©2025. Jorge Bengoa Pinedo

Esta obra está licenciada bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 3.0 Unported (CC BY-NC-ND 3.0). Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es> o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, EE.UU.

Todas las opiniones aquí expresadas son del autor, y no reflejan necesariamente las opiniones de la Universidad Politécnica de Madrid.

Título: Métodos paramétricos y redes neuronales recurrentes en el diagnóstico y pronóstico de la fase preclínica de la enfermedad de Alzheimer

Autor: Jorge Bengoa Pinedo

Tutor: Carlos Platero Dueñas

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de de ... en, en la Escuela Técnica Superior de Ingeniería y Diseño Industrial de la Universidad Politécnica de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A Carlos Platero por haber sido un tutor extraordinario. Por compartir conmigo su experiencia y llevarme a hombros de gigantes. Por abrirme la puerta a tantas herramientas y recursos que han ampliado enormemente las posibilidades de este trabajo. Por todas las tutorías que, enriquecidas por sus reflexiones, me han ayudado a no perder el rumbo. Por su tiempo, dedicación e implicación constante. Porque sin su ayuda este proyecto no habría sido posible tal como ha llegado a ser. Cualquier agradecimiento se queda corto.

A mis padres y mi hermana por el apoyo constante e incondicional que he recibido. Por su paciencia conmigo y por compartir sus opiniones en las situaciones más complicadas. Por ayudarme a seguir adelante y facilitarme el camino.

A todos mis amigos y compañeros de universidad. Especialmente a Luismi que, a pesar de no haber participado en este proyecto, ha estado ahí en tantos otros. A Bogurad por haberme dado el privilegio de escuchar sus reflexiones. A Fernando por contar siempre conmigo y por todas las tardes (y noches) de desconexión.

A Morna, porque no todo lo que cuenta se recoge en estas líneas.

A Cata por acompañarme, amenizar y apoyarme en este trayecto.

A quienes, involuntariamente, hubiera olvidado mencionar, siéntanse como si en estas palabras vieran su nombre.

Resumen

La enfermedad de Alzheimer es una enfermedad neurodegenerativa de carácter progresivo e irreversible que afecta a la memoria y funciones cognitivas de los pacientes. No solo causa un sinnúmero de cargas psicológicas y emocionales, sino que también supone una importante carga financiera sobre el sistema sanitario. Actualmente, solamente puede controlarse, pero no curarse. De la necesidad de una detección temprana surge la motivación de este proyecto. Teniendo esto en mente, se desarrollan modelos de progresión de la enfermedad que reconstruyen las trayectorias de los pacientes, estiman su diagnóstico y predicen su evolución. Los marcadores empleados, de tipo neuropsicológico, facilitan su implantación en la práctica clínica, reduciendo tiempos y costes.

Palabras clave: Enfermedad de Alzheimer, Fase preclínica, Diagnóstico, Pronóstico, Modelos de progresión de la enfermedad, Modelos paramétricos, Redes neuronales recurrentes, Computación de alto rendimiento.

Abstract

Alzheimer's disease is a progressive and irreversible neurodegenerative disease that affects patients' memory and cognitive functions. It not only causes a myriad of psychological and emotional burdens, but also imposes a significant financial burden on the healthcare system. Currently, it can only be controlled, but not cured. From the need for early detection arises the motivation for this project. With this in mind, disease progression models are developed to reconstruct patients' trajectories, estimate their diagnosis and predict their evolution. The use of neuropsychological markers facilitates implementation in clinical practice, reducing both time and costs.

Keywords: Alzheimer's disease, Preclinical phase, Diagnosis, Prognosis, Disease progression modelling, Parametric models, Recurrent neural networks, High performance computing.

Índice general

Agradecimientos	III
Resumen	IV
Abstract	V
Índice	IX
1. Introducción	1
1.1. La enfermedad de Alzheimer	2
1.1.1. Manifestaciones clínicas y deterioro cognitivo leve	2
1.1.2. Diagnóstico e intervención terapéutica	3
1.2. Impacto socioeconómico	4
1.3. Contexto de la neurotecnología e implicaciones éticas	6
1.4. Objetivos	7
1.5. Estructura del documento	7
2. Estado del arte	9
2.1. NIA-AA Research Framework: Toward a biological definition of Alzheimer’s disease	9
2.2. Detectable Neuropsychological Differences in Early Preclinical Alzheimer’s Disease: A Meta-Analysis	11
2.2.1. Sujetos A+ versus A-	12
2.2.2. Sujetos A+N+ versus A+N-	12
2.2.3. Resultados	12
2.3. Estimating long-term multivariate progression from short-term data	12
2.3.1. Introducción	12
2.3.2. Modelo y algoritmo	13
2.3.3. Resultados	14
2.4. Learning Multimodal Digital Models of Disease Progression from Longitudinal Data: Methods & Algorithms for the Description, Prediction and Simulation of Alzheimer’s Disease Progression	16
2.4.1. Geometría de Riemann	16
2.4.2. Modelos de progresión de la enfermedad	17
2.4.3. Métodos	18
2.5. Robust parametric modeling of Alzheimer’s disease progression	20
2.5.1. Métodos	20
2.5.2. Resultados	23
2.6. Temporal modeling and AT profiles in the early phase of Alzheimer’s disease	23
2.6.1. Métodos	24

2.6.2.	Resultados	25
2.7.	Predicting Alzheimer’s disease progression using deep recurrent neural networks	27
2.7.1.	Métodos	27
2.7.2.	Resultados	29
2.8.	CARRNN: A Continuous Autoregressive Recurrent Neural Network for Deep Representation Learning From Sporadic Temporal Data	30
2.8.1.	Métodos	31
2.8.2.	Resultados	33
2.9.	Deep recurrent model for individualized prediction of Alzheimer’s disease progression	34
2.9.1.	Métodos	34
2.9.2.	Resultados	37
2.10.	Multi-modal sequence learning for Alzheimer’s disease progression prediction with incomplete variable-length longitudinal data	38
2.10.1.	Métodos	39
2.10.2.	Resultados	42
2.11.	Aportaciones al trabajo	44
2.11.1.	NIA-AA Research Framework: Toward a biological definition of Alzheimer’s disease	44
2.11.2.	Detectable Neuropsychological Differences in Early Preclinical Alzheimer’s Disease: A Meta-Analysis	44
2.11.3.	Estimating long-term multivariate progression from short-term data	44
2.11.4.	Learning Multimodal Digital Models of Disease Progression from Longitudinal Data: Methods & Algorithms for the Description, Prediction and Simulation of Alzheimer’s Disease Progression	44
2.11.5.	Robust parametric modeling of Alzheimer’s disease progression	44
2.11.6.	Temporal modeling and AT profiles in the early phase of Alzheimer’s disease	44
2.11.7.	Predicting Alzheimer’s disease progression using deep recurrent neural networks	44
2.11.8.	CARRNN: A Continuous Autoregressive Recurrent Neural Network for Deep Representation Learning From Sporadic Temporal Data	45
2.11.9.	Deep recurrent model for individualized prediction of Alzheimer’s disease progression	45
2.11.10.	Multi-modal sequence learning for Alzheimer’s disease progression prediction with incomplete variable-length longitudinal	45
3.	Materiales	46
3.1.	Marcadores ADNI	46
3.1.1.	NM	46
3.1.2.	CSF	47
3.1.3.	T1-MRI	48
3.1.4.	PET	49
3.1.5.	Marcadores genéticos	49
3.2.	Programas utilizados	49
3.2.1.	PuTTY	49
3.2.2.	Xming	49
3.2.3.	FileZilla	50
3.2.4.	MATLAB	50

3.2.5.	R	50
3.2.6.	Python	50
3.2.7.	Anaconda	51
3.2.8.	tmux	51
3.2.9.	Magerit-3	52
3.2.10.	LaTeX	56
4.	Métodos	57
4.1.	Obtención de los datos	57
4.2.	Preprocesamiento de los datos	57
4.3.	Población de estudio	58
4.4.	Determinación del tiempo de conversión clínico	58
4.5.	Caracterización de la población de estudio	59
4.6.	Métodos paramétricos de progresión de la enfermedad	60
4.6.1.	GRACE	61
4.6.2.	Leaspy	64
4.6.3.	Reducción del problema multidimensional a unidimensional	64
4.7.	Métodos no paramétricos de progresión de la enfermedad	65
4.8.	Análisis masivo	66
4.8.1.	Medidas de calidad	70
4.8.2.	Partición de los datos	72
4.9.	Robustez frente a valores faltantes	72
4.10.	Análisis frente al número de visitas	73
4.11.	Pronóstico	73
4.11.1.	Modelos de progresión de la enfermedad	74
4.11.2.	Predicción constante	75
4.11.3.	Partición de los datos	76
4.11.4.	Medidas de calidad	76
4.12.	Historia natural	77
4.12.1.	Leaspy y GRACE	78
4.12.2.	RPDPM	78
4.13.	Arquitectura maestro-esclavo: paralelización con MPI	79
4.13.1.	Problemas de memoria en Tensorflow	80
5.	Resultados	82
5.1.	Caracterización de la población de estudio	82
5.2.	Clasificación de la población	89
5.3.	Análisis masivo	90
5.3.1.	Leaspy	91
5.3.2.	GRACE	91
5.3.3.	RPDPM	91
5.3.4.	LSTM	92
5.4.	DPS y tiempos de conversión	109
5.5.	Robustez frente a valores faltantes	111
5.6.	Análisis frente al número de visitas	120
5.7.	Pronóstico	121
5.8.	Historia natural	125
5.9.	Visualización de trayectorias	136

6. Discusiones	138
6.1. Caracterización de la población de estudio	138
6.2. Análisis masivo	139
6.3. DPS y tiempos de conversión	140
6.4. Robustez frente a valores faltantes	140
6.5. Análisis frente al número de visitas	141
6.6. Pronóstico	141
6.7. Historia natural	142
7. Conclusiones	144
7.1. Conclusión	144
7.2. Limitaciones	145
7.3. Desarrollos futuros	145
A. Estadística	147
A.1. Inferencia estadística	147
A.1.1. Contrastes de hipótesis	147
A.1.2. Prueba de chi cuadrado	148
A.1.3. ANOVA (1 factor)	148
A.2. Modelo Lineal de Efectos Mixtos (LME)	150
B. Clasificación	152
B.1. Introducción	152
B.2. Curva ROC y AUC	152
B.3. Clasificación multiclase	153
B.4. Clasificador de Bayes	154
B.4.1. KDE	154
C. Paralelización de tareas	156
C.1. Introducción	156
C.2. Paralelización en R	156
C.2.1. Paralelización en Windows	156
C.2.2. Paralelización en Linux	157
C.3. Paralelización en Python	158
C.3.1. Paralelización en Windows	158
C.3.2. Paralelización en Linux	159
C.4. Evaluación de la paralelización en Linux	159
C.5. Monitoreo del uso de RAM en Linux	160
C.6. Aceleración de tareas con CUDA/GPU	162
C.6.1. Aceleración en Magerit-3	163
D. Originalidad del documento	166
Lista de Acrónimos	167
Bibliografía	168

Índice de figuras

1.1.	Comparación entre un cerebro sano y otro afectado por la enfermedad de Alzheimer	3
1.2.	Defunciones por Alzheimer en España por sexo	5
2.1.	Perfiles AT(N) y categorías	10
2.2.	Combinación estado cognitivo con perfil AT(N)	11
2.3.	Panel de biomarcadores de ADNI	13
2.4.	Tabla de percentiles de ADNI	15
2.5.	Trayectorias a largo plazo de sujetos con niveles anormales de $A\beta$ (A) y de $APOE \epsilon 4$ (B)	15
2.6.	Subespacio del espacio \mathbb{R}^N correspondiente a todas las posibles imágenes del crecimiento de un niño	16
2.7.	Descripción geométrica del modelo donde cada observación $y_{ij} \in \mathbb{R}^N$ pertenece a la variedad Riemanniana \mathbf{M}	17
2.8.	Transformación de la edad a una escala común, DPS	21
2.9.	Probabilidad estimada del DPS según el estado de los sujetos (CN, MCI, AD)	24
2.10.	Diagrama de flujo que proporciona una visión general de la interacción entre los tres componentes principales que intervienen en el estudio: (a) modelos predictivos categóricos, (b) modelos de progresión de la enfermedad, (c) perfiles AT	25
2.11.	Trayectorias a largo plazo de los marcadores en la etapa temprana de AD, comparando sujetos A+ y A- al inicio del estudio	26
2.12.	Arquitectura de <i>minimalRNN</i>	28
2.13.	Resultados del AUC, BCA, y MAE del ADAS13 y del volumen ventricular de los modelos <i>minimalRNN-MF</i> , LSS-MF, LSTM-MF, predicción constante, y SVM (1 visita)	30
2.14.	Arquitecturas de los modelos CAR-LSTM, CAR-GRU y CAR-RNN	32
2.15.	Series temporales con eventos asíncronos y datos separados por intervalos de tiempo irregulares, y estructura CARRNN propuesta para modelar los datos agrupados	33
2.16.	Entrenamiento con valores faltantes utilizando un proceso ponderado de propagación hacia delante (<i>feedforward</i>) y retropropagación (<i>backpropagation</i>)	33
2.17.	(a) Arquitectura del modelo propuesto por Jung et al. combinando los módulos de imputación, codificación y predicción. (b) Operaciones para considerar las relaciones temporales y entre características en la imputación de valores faltantes. (c) Variante de la unidad LSTM estándar incluyendo γ y $\mathbf{m}_{x,t}$	35
2.18.	Representación del modelo propuesto por Xu et al.	39

2.19. Representación del módulo de fusión de modalidades propuesto por Xu et al.	40
2.20. Representación del módulo de aprendizaje de secuencias propuesto por Xu et al.	41
2.21. Representación del módulo de imputación propuesto por Xu et al.	42
3.1. Logo de PuTTY	49
3.2. Logo de Xming	50
3.3. Logo de FileZilla	50
3.4. Logo de MATLAB	50
3.5. Logos de R y RStudio	51
3.6. Logos de Python y Spyder	51
3.7. Logo de Anaconda	51
3.8. Logo de tmux	52
3.9. Logo de CesViMa	53
3.10. Logos de L ^A T _E X, MiKTeX y TeXstudio	56
4.1. Descarga de datos de la población ADNIMERGE	57
5.1. Histogramas con los tiempos de censura de sujetos sCU con medidas NM, y NM + CSF	89
5.2. Histogramas con los tiempos de conversión de sujetos pCU con medidas NM, y NM + CSF	89
5.3. Histogramas con los tiempos de censura de sujetos sSMC con medidas NM, y NM + CSF	90
5.4. Histogramas con los tiempos de conversión de sujetos pSMC con medidas NM, y NM + CSF	90
5.5. Histogramas con los tiempos de conversión estimados por Leaspy, GRACE y RPDPM para la población CU, SMC y sMCI	110
5.6. Histogramas con los tiempos de conversión estimados por Leaspy, GRACE y RPDPM para la población sCU, pCU, sSMC, pSMC y sMCI	111
5.7. Gráfico de dispersión con tiempo de conversión real frente a tiempos estimados por Leaspy, GRACE y RPDPM para la población pCU y pSMC	112
5.8. Gráfico de dispersión de la población pCU y pSMC con las estimaciones de tiempos de conversión entre los algoritmos propuestos: GRACE - Leaspy, GRACE - RPDPM y RPDPM - Leaspy	113
5.9. Diagramas de barras de las medidas de calidad frente al número de visitas según Leaspy, GRACE y RPDPM	121
5.10. Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente al número de visitas según Leaspy, GRACE y RPDPM	122
5.11. Diagramas de barras del AUC, BCA, %sCU, %sMCI, %pCU y número de sujetos frente a los años futuros según la predicción constante, Leaspy, GRACE y RPDPM	124
5.12. Diagramas de barras del AUC, BCA, %sCU, %sMCI, %pCU y número de sujetos frente a los años futuros según la predicción constante, Leaspy y LSTM	125
5.13. Diagramas de barras del %pCU, la correlación con la edad y con la reserva frente a los años futuros según Leaspy, GRACE y RPDPM	126
5.14. Diagramas de barras del %pCU, la correlación con la edad y con la reserva frente a los años futuros según Leaspy y LSTM	127

5.15. Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente a los años futuros según la predicción constante, Leaspy, GRACE y RPDPM 128

5.16. Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente a los años futuros según la predicción constante, Leaspy y LSTM 129

5.17. Histogramas del número de sujetos frente al tiempo de conversión según GRACE 130

5.18. Histogramas del número de sujetos frente al tiempo de conversión según RPDPM 130

5.19. Histogramas del número de sujetos frente al tiempo de conversión según Leaspy 131

5.20. Histogramas del número de sujetos frente al tiempo de conversión según LSTM 131

5.21. Gráficos de dispersión de los tiempos de conversión estimados por Leaspy (AC), RPDPM (MC) y GRACE (MC) de sujetos sCU, pCU y sMCI 132

5.22. Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTO-TAL, ABETA, NHV propuesta por GRACE 133

5.23. Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTO-TAL, ABETA, NHV propuesta por Leaspy 133

5.24. Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTO-TAL, ABETA, NHV propuesta por RPDPM 134

5.25. Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTO-TAL, ABETA, NHV propuesta por LSTM 134

5.26. Visualización de las trayectorias propuestas por Leaspy (AC) para los marcadores ADAS13, CDRSB en escala original del sujeto con el identificador 8 136

5.27. Visualización de las trayectorias propuestas por LSTM (R_LFC) para los marcadores RAVLT learning, FAQ, CDRSB en escala original del sujeto con el identificador 8 137

5.28. Visualización de las trayectorias propuestas por Leaspy (AC) y LSTM (R_LFC) para los marcadores ADAS13, CDRSB en escala percentil y RAVLT learning, FAQ, CDRSB en escala estandarizada del sujeto con el identificador 8 137

B.1. Curva ROC para distintos clasificadores 153

B.2. Funciones de densidad estimadas mediante KDE con diferentes anchos de banda 155

C.1. Posible resultado de la ejecución de *htop*. 160

C.2. Posibles resultados de la ejecución de `vmstat -SM intervalo_segundos número_intervalos > $SCRATCH/memoryusage.$SLURM_JOB_ID.out &` . . 162

D.1. Resumen del informe de similitud de Turnitin. 166

Índice de tablas

1.1. Escala que muestra las diferentes etapas del deterioro cognitivo que sufre un paciente con la enfermedad de Alzheimer	4
2.1. Métrica $g_p^k(u, v)$, ecuación de la geodésica $\gamma_k(t)$ y de la trayectoria individual $\eta_k^{w_i}(\psi_i(t_{ij}))$ para distintos tipos de evoluciones temporales propuestas: rectas paralelas (a), rectas (b), logísticas paralelas (c), logísticas (d), exponenciales paralelas (e), exponenciales (f)	19
2.2. Funciones logísticas consideradas para ajustar las trayectorias de los marcadores	21
2.3. Estimadores M considerados para la regresión robusta	22
2.4. Puntuaciones para predecir conversores rápidos de CU a MCI	26
2.5. Vectores de características de los DPM propuestos con sus correlaciones y puntuaciones	27
2.6. Espacio de búsqueda de hiperparámetros para <i>minimalRNN</i> , LSS y LSTM	29
2.7. Espacio de búsqueda de hiperparámetros para SVM/SVR	29
2.8. Espacio de búsqueda de hiperparámetros de los distintos métodos utilizados por Xu et al.	43
5.1. Características demográficas y clínicas iniciales de la población CN, SMC, MCI y AD con marcadores NM	84
5.2. Características demográficas y clínicas iniciales de la población CN, SMC, MCI y AD con marcadores NM y CSF	85
5.3. Características demográficas y clínicas iniciales de la población sCU, pCU, sSMC y pSMC con marcadores NM	86
5.4. Características demográficas y clínicas iniciales de la población sCU, pCU, sSMC y pSMC con marcadores NM y CSF	87
5.5. Características demográficas y clínicas de la población de entrenamiento sCU, pCU, sMCI con marcadores NM	88
5.6. Mejores vectores de características para el algoritmo Leaspy	94
5.7. Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo Leaspy	95
5.8. Mejores vectores de características para el algoritmo GRACE	96
5.9. Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo GRACE	97
5.10. Mejores vectores de características para el algoritmo RPDPM	98
5.11. Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo RPDPM	99
5.12. Mejores vectores de características para el algoritmo LSTM	100
5.13. Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo LSTM	101

5.14. MAE para los mejores vectores de características entrenados con Leaspy . .	102
5.15. MAE para los mejores vectores de características de dimensión reducida entrenados con Leaspy	103
5.16. MAE para los mejores vectores de características entrenados con GRACE .	104
5.17. MAE para los mejores vectores de características de dimensión reducida entrenados con GRACE	105
5.18. MAE para los mejores vectores de características entrenados con RPDPM .	105
5.19. MAE para los mejores vectores de características de dimensión reducida entrenados con RPDPM	106
5.20. MAE para los mejores vectores de características entrenados con LSTM . .	107
5.21. MAE para los mejores vectores de características de dimensión reducida entrenados con LSTM	108
5.22. Medidas de calidad para el vector ADAS13, CDRSB entrenado con Leaspy al introducir valores NaN	114
5.23. Medidas de calidad para el vector MMSE, CDRSB entrenado con GRACE al introducir valores NaN	115
5.24. Medidas de calidad para el vector MMSE, CDRSB entrenado con RPDPM al introducir valores NaN	116
5.25. Medidas de calidad para el vector RAVLT_learning, FAQ, CDRSB entre- nado con LSTM al introducir valores NaN	117
5.26. MAE para el vector ADAS13, CDRSB entrenado con Leaspy al introducir valores NaN	118
5.27. MAE para el vector MMSE, CDRSB entrenado con GRACE al introducir valores NaN	118
5.28. MAE para el vector MMSE, CDRSB entrenado con RPDPM al introducir valores NaN	118
5.29. MAE para el vector RAVLT_learning, FAQ, CDRSB entrenado con LSTM al introducir valores NaN	119
5.30. Ecuaciones de regresión y coeficientes de correlación con la reserva entre algoritmos, por grupo clínico.	129
5.31. Valores medios de los grupos sCU, pCU y sMCI para las características ADAS13, MMSE, CDRSB, ABETA y NHV al inicio del estudio; valores de corte utilizados en la práctica clínica para el diagnóstico de MCI; valores de corte estimados por Leaspy, GRACE, RPDPM y LSTM.	135

Índice de algoritmos

1.	getADNIMERGE	58
2.	tonset	59
3.	categories_nm y categories_nm_csf	60
4.	convert_censure_nm y convert_censure_nm_csf	61
5.	Personalización de GRACE	63
6.	Preparación de los datos	64
7.	Alineación temporal	67
8.	Construcción de las secuencias	68
9.	Función de pérdida con ponderación del MAE y BCE	68
10.	multiGrace, multiLeaspy y multiLeaspy	71
11.	Reemplazamiento por NaN	73
12.	Análisis frente al número de visitas con GRACE / Leaspy	74
13.	Pronóstico con GRACE / Leaspy	75
14.	Historia natural con GRACE / Leaspy	79
15.	Arquitectura maestro-esclavo para la paralelización de tareas usando MPI .	81
16.	Revisión del algoritmo 15 para evitar acumulación de memoria por modelos consecutivos entrenados con Tensorflow	81

Índice de códigos

- C.1. Código `.sh` para lanzar un trabajo en R en Magerit-3. Se solicitan un total de 40 tareas distribuidas en 1 único nodo. El tiempo máximo de ejecución del trabajo es de 50 horas. 164
- C.2. Código `.sh` para lanzar un trabajo en python en Magerit-3. Se solicitan un total de 40 tareas distribuidas en 1 único nodo. El tiempo máximo de ejecución del trabajo es de 50 horas. 165

Capítulo 1

Introducción

El envejecimiento es un proceso universal, continuo e irreversible que provoca la pérdida de vitalidad, aumento de la vulnerabilidad y, eventualmente, la muerte. La epigenética es el factor más influyente, por lo que cada persona envejece de forma única.

Con la edad se produce una variación en la expresión de los genes (epigenética) que altera el funcionamiento de las células, dando lugar a estrés oxidativo¹. Como consecuencia, aparecen mutaciones en el ADN mitocondrial, que repercuten en la expresión y actividad de proteínas implicadas en la síntesis energética (ATP²). Este proceso afecta especialmente al cerebro³, ya que su consumo de oxígeno representa aproximadamente un 20 % del total. En conjunto, se genera una relación multidireccional entre estrés oxidativo, mal funcionamiento mitocondrial, incremento de la apoptosis⁴ y disminución de la autofagia⁵ que desembocan en neurodegeneración.

Las enfermedades neurodegenerativas son trastornos crónicos, hereditarios o adquiridos, que causan una disfunción del sistema nervioso central. Son el producto de la acumulación de proteínas mal plegadas que dan lugar a un daño oxidativo e inflamatorio que conduce a la muerte neuronal. Teniendo en cuenta que las neuronas son postmitóticas⁶, los daños se acumulan a lo largo de la vida del individuo. No solamente se produce una gran discapacidad intelectual y social⁷ como resultado, sino que también tienen una repercusión laboral y un alto gasto económico debido a la atención social y sanitaria precisada.

De acuerdo con el NINDSS (*National Institute of Neurological Disorder and Stroke Study*), hay más de 600 enfermedades neurodegenerativas. Entre ellas destacan la enfermedad de Alzheimer, la enfermedad de Parkinson, la enfermedad de Huntington y la esclerosis lateral amiotrófica.

¹El estrés oxidativo surge cuando los neutralizantes (*scavengers*) no son suficientes para neutralizar los radicales libres derivados de la actividad mitocondrial. Los radicales libres son especies químicas con al menos 1 electrón desapareado en su última órbita. Estos reaccionan con otras moléculas, cediendo o arrebatando electrones, y convirtiendo a las moléculas en nuevos radicales libres. Este proceso en cadena, que se detiene cuando se encuentran 2 radicales libres, origina una gran cantidad de sustancias inestables y tóxicas.

²La mayor parte del oxígeno que llega a las células neuronales se utiliza para sintetizar la molécula ATP, necesaria para mantener el gradiente iónico de la célula. El mantenimiento de este gradiente es absolutamente necesario para alcanzar una correcta transmisión del impulso nervioso.

³El cerebro está formado por neuronas y células gliales.

⁴La apoptosis (muerte celular) es un proceso ordenado por el que el contenido celular se empaqueta en compartimentos para que las células del sistema inmune los recojan. Su finalidad es mantener/adaptar las distintas estructuras (p.ej. las membranas que tienen los dedos cuando están dentro del vientre desaparecen por apoptosis).

⁵La autofagia es un proceso de reciclaje por el cual los componentes de citoplasma forman autofagosomas y son entregados a los lisosomas (centros de reciclaje). Su finalidad es mantener la homeostasis. Alteraciones en este mecanismo dan lugar al acúmulo de proteínas, provocando neurodegeneración.

⁶Las células postmitóticas son aquellas que han alcanzado su estado final y no pueden dividirse más

⁷Y, por lo tanto, dependencia y disminución de la calidad de vida.

1.1. La enfermedad de Alzheimer

La enfermedad de Alzheimer (AD, *Alzheimer's Disease*) es un trastorno progresivo e irreversible que ocasiona la pérdida de memoria, capacidad de pensamiento y de llevar a cabo actividades cotidianas. Las cifras de prevalencia se doblan cada cinco años, desde el 1-2% a los 65-70 años hasta el 30% en personas mayores de 85 años. La supervivencia media se establece entre 8 y 12 años, aunque se registran casos de más de 20 años de evolución.

Se documentan numerosos factores de riesgo:

- Sexo: en edades avanzadas, presenta una mayor incidencia en mujeres.
- Bajo nivel de escolarización.
- Factores de riesgo vascular: hipertensión arterial y colesterol total elevado en edades medias. Niveles bajos de tensión arterial y del colesterol total en edades avanzadas.
- Anemia y alteraciones de los niveles de la hormona estimulante del tiroides o de hormonas sexuales.
- Ser fumador activo: se duplica el riesgo de padecer demencia.
- Obesidad e infrapeso.
- Depresión: se estima que aumenta en dos veces el riesgo.
- Factores genéticos: aunque la mayoría de los casos son de tipo esporádico (no hereditario), el genotipo apolipoproteína E (ApoE), en especial la forma alélica $\epsilon 4$, se relaciona con el enfermedad de Alzheimer.

En pacientes con AD se aprecia una disminución del volumen cerebral (especialmente en el lóbulo temporal), la corteza se vuelve más delgada y se produce una pérdida de sinapsis. Además, aparecen depósitos de $A\beta$ (placas seniles) extracelulares y almacenamiento intracelulares de proteína tau anormalmente fosforilada (ovillos neurofibrilares). En el entorno de las placas amiloides se observa una serie de elementos ligados a la existencia de procesos inflamatorios y estrés oxidativo⁸. La figura 1.1 muestra una comparativa entre un cerebro sano y otro afectado por la enfermedad de Alzheimer.

1.1.1. Manifestaciones clínicas y deterioro cognitivo leve

La demencia es un proceso progresivo en varias etapas:

1. Amnesia de evocación (mismas preguntas, incapacidad de aprender).
2. Afasia (pérdida del lenguaje).
3. Desorientación espacial, temporal y personal.

⁸Se produce inflamación como defensa contra infecciones, con la finalidad de restablecer los tejidos. En el caso de la enfermedad de Alzheimer, las células gliales rodean a las placas amiloides, generando un entorno de inflamación. Dicho entorno promueve la actividad de la enzima BACE (ligada a la formación de $A\beta$). Las placas amiloides producen estrés oxidativo (neurotoxicidad), lo que provoca neuroinflamación. De esta forma, se forma un círculo vicioso que convierte un proceso de defensa en un proceso de destrucción neuronal.

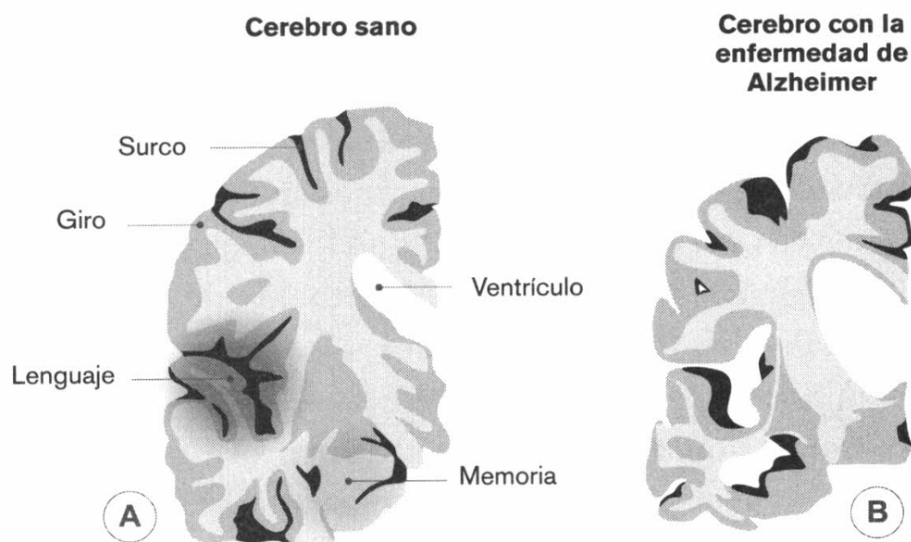


Figura 1.1: Comparación entre un cerebro sano (izquierda) y otro afectado por la enfermedad de Alzheimer (derecha). Fuente: P.G. Gregorio [33]

La disfunción ejecutiva origina la desorganización en tareas cotidianas. El paciente es incapaz de hacer cosas de forma simultánea, pierde capacidad de juicio crítico y razonamiento, y presenta déficits en el pensamiento abstracto y pérdida de iniciativa.

Además, aparecen síntomas conductuales como apatía, agresividad o deambulación errática (caminar sin rumbo). Se producen alucinaciones visuales y alteraciones del ritmo vigilia-sueño.

El término deterioro cognitivo leve surge de la necesidad de definir un estado intermedio entre la normalidad y la demencia. Se caracteriza por quejas de memoria del paciente/-familiares, declive en la función cognitiva, y dificultades en la realización de actividades complejas. Puede venir acompañado de desorientación en lugares habituales, dificultades para encontrar una palabra, depresión, irritabilidad, ansiedad, agresividad o apatía. Es preciso enfatizar que no se trata de demencia.

En general, es conveniente ubicar al paciente en un estado evolutivo determinado para una adecuada programación del tratamiento. A estos efectos, la escala más utilizada es la *General Deterioration Scale* (GDS) de Reisberg (figura 1.1):

1.1.2. Diagnóstico e intervención terapéutica

El diagnóstico se basa en la información aportada por el paciente y sus cuidadores. Se complementa con un sistema de valoración cognitivo, conductual, funcional. Además, se apoya en biomarcadores ($A\beta$ y proteína tau fosforilada en el líquido cefalorraquídeo, LCR). En los últimos años han aparecido técnicas de mayor especificidad (p.ej PET-FDG).

La intervención puede ser:

- Farmacológica:

Estadio GDS	Síntomas cognitivos
GDS1: Ausencia de deterioro	
GDS2: Disminución cognitiva leve	Quejas subjetivas de defectos de memoria. No hay defectos objetivos en el trabajo o en situaciones sociales.
GDS3: Deterioro cognitivo leve	Dificultad para evocar palabras/nombres. Recuerda poco de lo que ha leído. Puede perderse en un lugar no familiar. Hay negación de los defectos.
GDS4: Deterioro cognitivo moderado	Déficit en el recuerdo. Conocimiento disminuido de acontecimientos actuales. Disminución de la capacidad de viajar. Disminución de las muestras de afecto. La negación es el mecanismo de defensa.
GDS5: Demencia leve	Incapacidad para recordar aspectos como direcciones, teléfonos, nombre de familiares. Desorientación temporoespacial. No puede permanecer mucho tiempo sin asistencia. Dificultad para elegir vestido.
GDS6: Demencia moderada	Desconocimiento de los acontecimientos recientes, pero cierto reconocimiento de la vida pasada. Desconocimiento del entorno (año, mes). Requiere asistencia para realizar actividades cotidianas.
GDS7: Demencia grave	Ideas delirantes, obsesiones, apatía. Se van perdiendo las habilidades básicas psicomotoras como andar. Incontinencia urinaria, requiere asistencia para su higiene y alimentación. Pérdida de todas las capacidades verbales.

Tabla 1.1: Escala que muestra las diferentes etapas del deterioro cognitivo que sufre un paciente con la enfermedad de Alzheimer. Fuente: P.G. Gregorio [33]

- Paliar la degeneración neuronal: antiinflamatorios, disminución de $A\beta$...
- Potenciación de las neuronas no afectadas.
- Potenciación del metabolismo cerebral y contrarrestar déficits de neurotransmisores.
- ...
- No farmacológica:
 - Terapia de reminiscencia: usa elementos del pasado del paciente (p.ej. fotos).
 - Estimulación magnética transcraneal repetitiva: se provoca una corriente en determinadas áreas.
 - Programas de orientación a la realidad: informa al paciente sobre su entorno físico y social.
 - ...

1.2. Impacto socioeconómico

Según la Organización Mundial de la Salud (OMS), a 7 de diciembre de 2017, el número de personas que padecían demencia se situaba en 50 millones. Se espera que dicha cifra se

triplique para 2050, alcanzando los 152 millones de casos. El costo anual de la demencia se calcula en US\$ 818 000 millones, lo que representa un 1 % del producto interior bruto mundial. Está previsto que este valor ascienda a US\$ 2 billones para 2030, lo cual podría socavar el desarrollo social y económico, y desbordar los servicios sociales y de salud. La enfermedad de Alzheimer es la forma más frecuente de demencia, concentrando entre un 60 % y un 70 % de los casos [41].

De acuerdo con la Sociedad Española de Neurología (SEN), a 19 de septiembre de 2019, 800000 personas en España padecen la enfermedad de Alzheimer. Los diagnósticos se sitúan entre un 3 % y un 4 % de la población entre 75 y 79 años, llegando hasta un 34 % en mayores de 85 años. El 15 % de la población por encima de 65 años padece deterioro cognitivo leve, debido en un 50 % de los casos a la enfermedad de Alzheimer. Cada año se diagnostican en España 40000 nuevos casos de Alzheimer. Sin embargo, se estima que el 80 % de los casos que aun son leves están sin diagnosticar, y que entre el 30 % y 40 % de los casos totales también lo estarían [48].

En España, la SEN calcula que el coste medio de un paciente con Alzheimer oscila entre 17100€ y 28200€. Cifras que pueden llegar hasta los 41700€ en los casos graves, en los que una parte de los pacientes requieren institucionalización. Teniendo en cuenta todos los niveles de gravedad, el costes total estimado en España del tratamiento del Alzheimer en pacientes mayores de 65 años es de unos 10000 millones de euros, lo que representa el 1.5 % del producto interior bruto nacional [48].

El Instituto Nacional de Estadística (INE) recogió en 2020 un total de 15571 muertes atribuidas al Alzheimer en España aquel año. La tendencia sigue un aumento sostenido, con una incidencia mayor en mujeres (figura 1.2). Aquel año las muertes por Alzheimer representaron el 3.15 % del total de defunciones en España, superando el 1 % de las registradas en mayores de 70 años y el 4 % entre los 85 y 94 años. Estos porcentajes fueron inferiores a los de 2019 (3.5 % de las muertes totales) debido a la pandemia [30].

Defunciones por Alzheimer en España por sexo

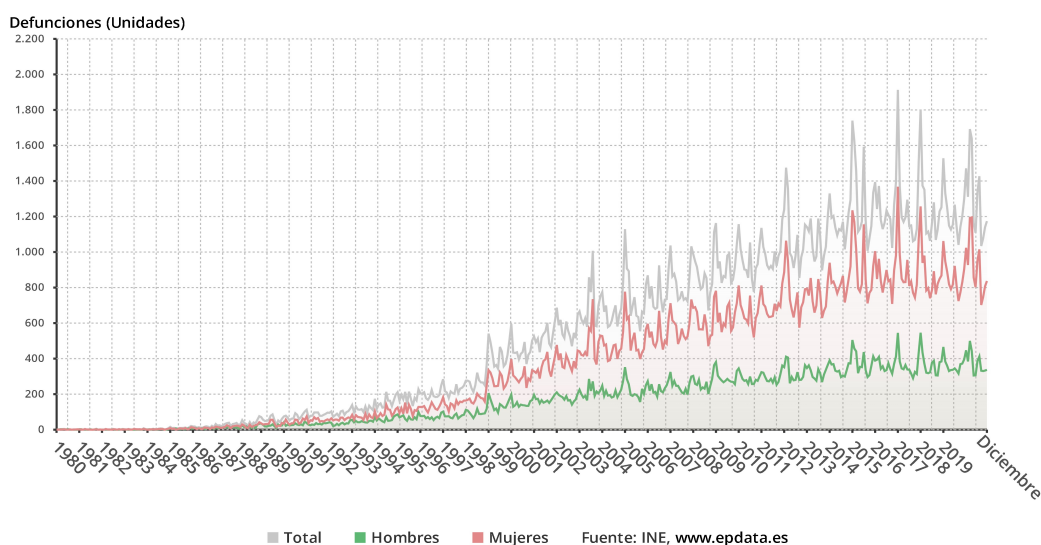


Figura 1.2: Defunciones por Alzheimer en España por sexo. Fuente: INE [30]

1.3. Contexto de la neurotecnología e implicaciones éticas

Las neurotecnologías permiten una conexión directa entre un dispositivo y el cerebro, con objeto de registrar o modificar la actividad nerviosa. La investigación en este campo ha permitido profundizar en el conocimiento de los mecanismos de la memoria, el lenguaje, el comportamiento y la coordinación cerebral. No obstante, se abre un debate que aborda cuestiones legales, éticas y morales sobre su impacto en la sociedad [40].

Los avances en diversas disciplinas, destacándose la inteligencia artificial y el aprendizaje automático, han permitido el desarrollo de interfaces cerebro-ordenador capaces de ofrecer soluciones que parecían impensables:

- **Neuroprótesis y neurorrehabilitación:** las neuroprótesis son dispositivos implantables que pueden ayudar a recuperar funciones motoras y sensoriales en personas que han sufrido lesiones graves o enfermedades neurodegenerativas. Interpretan las señales eléctricas del cerebro y las traducen en movimientos o sensaciones. Como consecuencia, se han documentado casos de pacientes tetraplégicos que han conseguido recuperar la movilidad en las piernas.
- **Decodificación del lenguaje, pensamiento visual y estados emocionales:** se ha demostrado la capacidad de interpretación de señales cerebrales, resultando en que personas con parálisis puedan navegar por internet y tener acceso a servicios tan solo pensando una palabra. Por otro lado, la decodificación de estados emocionales a partir de estas señales podría tener especial relevancia en el tratamiento de enfermedades como la depresión.
- **Aplicaciones no clínicas:** por ejemplo, el monitoreo cognitivo, la comunicación y el control de dispositivos. También se incluye el neuromarketing. La inversión en esta clase de aplicaciones se ha multiplicado por 21 desde 2012, sobrepasando los 30 mil millones de euros. Se espera que en el medio plazo (5-10 años) se logren tecnologías que mejoren nuestra mente, capacidad cognitiva y estado de consciencia.

Asociadas al desarrollo de las neurotecnologías surgen importantes cuestiones éticas. El principal desafío es la privacidad mental, dado el potencial riesgo de extraer neurodatos confidenciales sin el consentimiento del individuo. A esto se suman los cambios producidos en las personas que pueden derivarse del uso de algunas neurotecnologías. Algunos pacientes han llegado a dudar de su capacidad de su identidad y capacidad de decisión, lo que supone una amenaza para el libre albedrío. Otro debate gira en torno a la mejora de las capacidades cognitivas, pues el acceso no equitativo a estas tecnologías podría contribuir a aumentar las desigualdades económicas, sociales, o culturales. Contar con el apoyo de la sociedad puede ser un elemento clave en el desarrollo de las neurotecnologías. En general, se acepta su uso para el ámbito médico; sin embargo, las aplicaciones en defensa o seguridad nacional provocan mayor desconfianza.

Para dar respuesta a los problemas que entraña el desarrollo de la neurociencia, se plantea la adecuación de los marcos legislativos para asegurar la protección de los neuroderechos de los ciudadanos. En el ámbito clínico, la nueva normativa exige que cualquier producto sanitario deba superar una evaluación rigurosa que demuestre sus beneficios clínicos, seguridad y eficacia. En cuanto a las aplicaciones sin finalidad médica, la comunidad científica propone categorizar los neurodatos como datos sanitarios sensibles. Además, dichos productos deberán ser sometidos a un análisis de riesgo exhaustivo, debiendo comunicarse sus advertencias, efectos secundarios y contraindicaciones al usuario.

A pesar de la legislación existente (p.ej. la Ley de Privacidad y Protección de Datos o la Ley de Inteligencia Artificial), la comunidad científica sugiere una revisión de la misma y ofrece algunas recomendaciones.

Este proyecto se sirve de técnicas de la inteligencia artificial y análisis de datos con el objetivo de lograr una detección temprana y pronóstico de la evolución de la enfermedad de Alzheimer. Consecuentemente, la aplicación de tratamientos médicos desde un inicio contribuirá a retrasar la aparición de los síntomas y mejorar la calidad de vida del paciente.

1.4. Objetivos

Este proyecto persigue los siguientes objetivos:

- Construcción de modelos de progresión de la fase preclínica de la enfermedad de Alzheimer capaces de estimar tanto las trayectorias poblacionales como individuales. Se explorarán diversos modelos paramétricos, así como redes neuronales recurrentes de tipo LSTM (*Long Short Term Memory*).
- Determinación de los marcadores que mejor capturan el progreso de la enfermedad para cada modelo. Los conjuntos resultantes deben tener una dimensión lo más reducida posible y ser de tipo neuropsicológico. De esta forma, se facilita su implantación en la práctica clínica, reduciendo tiempos y costes.
- Establecer un marco comparativo aplicable tanto a los modelos que se analizan en este estudio como a futuras propuestas. Se evaluarán en tareas de diagnóstico y pronóstico, robustez ante valores faltantes e influencia del número de visitas.
- Reconstrucción de la historia natural de los pacientes. Serán situados en la escala temporal de la enfermedad y se podrá obtener su trayectoria más allá del límite de sus visitas.
- Integración del código para su ejecución eficiente en entornos de cómputo intensivo.
- Documentación detallada de las herramientas, métodos y procedimientos empleados.

1.5. Estructura del documento

A continuación y para facilitar la lectura del documento, se detalla el contenido de cada capítulo:

- **Capítulo 1: Introducción.** Se explican conceptos básicos sobre la enfermedad de Alzheimer, el contexto de la neurotecnología y se presentan los objetivos del proyecto.
- **Capítulo 2: Estado del arte.** Se exponen los artículos en los que se basa el estudio.
- **Capítulo 3: Materiales.** Se indican las herramientas empleadas, así como su aplicación.
- **Capítulo 4: Métodos.** Se describe los métodos de obtención, preprocesado y explotación de los datos.
- **Capítulo 5: Resultados.** Se exponen los resultados obtenidos de la aplicación de los métodos.

- **Capítulo 6: Discusiones.** Se analizan los resultados y se comparan con otros estudios relacionados.
- **Capítulo 7: Conclusiones.** Se presentan las conclusiones alcanzadas, limitaciones y líneas de investigación futuras.
- **Apéndices.** Se proporciona información adicional que detalla algunos aspectos del proyecto.

Capítulo 2

Estado del arte

Los artículos en los que se apoya este proyecto se pueden dividir en tres categorías. En primer lugar, aquellos que aportan un marco teórico común en relación a la enfermedad de Alzheimer sobre el que trabajar [35], [29]. En segundo lugar, los que explican el fundamento matemático de los modelos paramétricos empleados (GRACE, Leaspy y RPDPM) [28], [38], [31], [47]. En tercer lugar, los que describen el funcionamiento de diversas propuestas de redes neuronales recurrentes para la descripción de la progresión de la enfermedad [39], [32], [36], [50]. Finalmente, se resumen las aportaciones de cada artículo al proyecto.

2.1. NIA-AA Research Framework: Toward a biological definition of Alzheimer's disease

El objetivo de este estudio es proporcionar un marco de trabajo común para los investigadores de la Enfermedad de Alzheimer (AD). De tal forma, los subsecuentes trabajos compartirán un mismo lenguaje y las conclusiones obtenidas serán comparables entre sí. Cabe decir que dicho marco se restringe a un ámbito de investigación y no debe usarse como criterio diagnóstico en la práctica clínica.

La AD se puede definir y clasificar en base tres (por ahora) biomarcadores:

- Placas $A\beta$ (A, específico): el individuo se encuentra en el espectro del Alzheimer. Determinado por PET Amiloide y CSF $A\beta_{42}$.
- Depósitos de tau neurofibrilar (T, específico): necesario junto a $A\beta$ para concluir AD. Determinado por CSF pTau y tau Pet.
- Neurodegeneración ((N), no específico): gravedad de la patología. Determinado por CSF tTau, PET-FDG y MRI.

De esta forma, se distinguen distintas categorías (figura 2.1) según el perfil AT(N). Un sujeto será incluido en la categoría AD si resulta positivo en A y T. A diferencia de la práctica pasada, esto es independiente de la existencia de síntomas. Es necesario enfatizar:

- Los biomarcadores son variables continuas. Sin embargo, podría resultar conveniente realizar una discretización por 1 ($A^+T^+(N)^+$) o varios ($A^2T^1(N)^0$) puntos de corte.
- El sistema AT(N) incluye biomarcadores CSF y por imágenes en cada uno de los grupos. Así se prevee la disponibilidad de solo uno o ambos tipos de pruebas. En el caso de usar ambos, los resultados deberían concordar en el largo plazo.

AT(N) profiles	Biomarker category	
A-T-(N)-	Normal AD biomarkers	
A+T-(N)-	Alzheimer's pathologic change	Alzheimer's continuum
A+T+(N)-	Alzheimer's disease	
A+T+(N)+	Alzheimer's disease	
A+T-(N)+	Alzheimer's and concomitant suspected non Alzheimer's pathologic change	
A-T+(N)-	Non-AD pathologic change	
A-T-(N)+	Non-AD pathologic change	
A-T+(N)+	Non-AD pathologic change	

Figura 2.1: Perfiles AT(N) y categorías. Fuente: Jack et al. [35]

Se propone también una clasificación adicional según el estado cognitivo del paciente en base a 2 criterios:

- Estado cognitivo sindrómico: válido para cualquier perfil AT(N). Los síntomas no pueden atribuirse directamente a la AD:
 - Sin deterioro cognitivo (CU): rendimiento cognitivo dentro del rango esperado.
 - Deterioro cognitivo leve (MCI): rendimiento cognitivo por debajo del rango esperado. Puede tener impacto en la realización de actividades más complejas.
 - Demencia: el deterioro cognitivo es progresivo y afecta a varios dominios neuroconductuales. Dependencia en la realización de actividades cotidianas. Puede dividirse a su vez en: leve, moderada, grave.

La clasificación combinaría el perfil AT(N) con el estado cognitivo: $A^+T^+(N)^+$ MCI (figura 2.2).

- Estado numérico clínico: válido para perfiles en el espectro del Alzheimer. Representa la evolución de la AD desde un estado inicial:
 1. Estado 1 (asintomático): no hay síntomas neuroconductuales ni de deterioro cognitivo.
 2. Estado 2 (deterioro cognitivo transicional): aparición de los primeros efectos clínicos detectables. No tiene impacto en la actividad diaria.
 3. Estado 3 (deterioro cognitivo leve): anormalidad en pruebas cognitivas. Menor eficiencia en la ejecución de tareas cotidianas.
 4. Estado 4 (demencia leve): el deterioro es sustancial y afecta a varios dominios neuroconductuales. Es necesaria ayuda ocasional con las actividades diarias.

		Cognitive stage		
		Cognitively Unimpaired	Mild Cognitive Impairment	Dementia
Biomarker Profile	$A^-T^-(N)^-$	normal AD biomarkers, cognitively unimpaired	normal AD biomarkers with MCI	normal AD biomarkers with dementia
	$A^+T^-(N)^-$	Preclinical Alzheimer's pathologic change	Alzheimer's pathologic change with MCI	Alzheimer's pathologic change with dementia
	$A^+T^+(N)^-$	Preclinical Alzheimer's disease	Alzheimer's disease with MCI(Prodromal AD)	Alzheimer's disease with dementia
	$A^+T^+(N)^+$	Alzheimer's and concomitant suspected non Alzheimer's pathologic change, cognitively unimpaired	Alzheimer's and concomitant suspected non Alzheimer's pathologic change with MCI	Alzheimer's and concomitant suspected non Alzheimer's pathologic change with dementia
	$A^-T^+(N)^-$	non-Alzheimer's pathologic change, cognitively unimpaired	non-Alzheimer's pathologic change with MCI	non-Alzheimer's pathologic change with dementia
	$A^-T^+(N)^+$	non-Alzheimer's pathologic change, cognitively unimpaired	non-Alzheimer's pathologic change with MCI	non-Alzheimer's pathologic change with dementia
	$A^+T^-(N)^+$	non-Alzheimer's pathologic change, cognitively unimpaired	non-Alzheimer's pathologic change with MCI	non-Alzheimer's pathologic change with dementia

Figura 2.2: Combinación estado cognitivo con perfil AT(N). Fuente: Jack et al. [35]

5. Estado 5 (demencia moderada): ejecución de actividades básicas afectada. Necesaria ayuda frecuente en las actividades diarias.
6. Estado 6 (demencia grave): completa dependencia con impacto funcional grave en la vida diaria.

La nomenclatura se puede simplificar estableciendo:

- $A^+T^-(N)^-$: a
- $A^+T^+(N)^-$: b
- $A^+T^+(N)^+$: c
- $A^+T^-(N)^+$: d

La clasificación combinaría el estado clínico con el perfil AT(N): 2d.

2.2. Detectable Neuropsychological Differences in Early Preclinical Alzheimer's Disease: A Meta-Analysis

La identificación de biomarcadores de la fase preclínica del AD es esencial para prevenir o intervenir de forma temprana. Actualmente, la presencia de patologías amiloide y tau son los mejores indicadores. Sin embargo, su relación con la pérdida de habilidades cognitivas no es tan evidente.

En este estudio, se lleva a cabo un meta-análisis para evaluar la influencia de las patologías amiloide y tau en la función cognitiva de sujetos clasificados como cognitivamente normales. Para ello, se recopilan numerosos estudios atendiendo a un estricto criterio de exclusión con el fin de obtener unos resultados lo menos sesgados posible. Algunos de estos criterios son:

- La edad media de los participantes tiene que ser mayor de 50 años.
- Los participantes deben ser cognitivamente normales, excluyendo incluso aquellos con deterioro cognitivo sutil (a medio camino entre CU y MCI).
- Los estudios no deben estar repetidos.
- El diseño de los estudios debe ser de alta calidad y no puede estar influenciado por fuentes de financiación.

De las 775 fuentes seleccionadas, únicamente 61 se incluyeron en el estudio.

Los marcadores de los distintos estudios se agrupan en 7 dominios cognitivos: función cognitiva global, memoria, lenguaje, visuoespacial, velocidad de procesamiento, memoria de trabajo, y función premórbida. Para cada dominio se comprueban las siguientes hipótesis:

- Existe diferencia en la función cognitiva de adultos CU con (A+) y sin (A-) patología amiloide. Si la diferencia es significativa, se hace un análisis post-hoc para comprobar hasta qué punto el resultado es dependiente del tipo de biomarcador utilizado (CSF o PET). También se evalúa la influencia de la edad en los resultados.
- Existe diferencia en la función cognitiva de adultos CU con (A+N+) y sin (A+N-) patología tau o neurodegeneración, entre los que presentan patología amiloide.

2.2.1. Sujetos A+ versus A-

Los resultados muestran que los sujetos A- obtuvieron mejores resultados en función cognitiva global, memoria, lenguaje, visuoespacial, velocidad de procesamiento, y memoria de trabajo. Los sujetos A+ obtuvieron puntuaciones más elevadas en la función premórbida. Después del análisis post-hoc se concluye que el lenguaje es dependiente del tipo de biomarcador, y que la memoria, el lenguaje y la velocidad de procesamiento dependen de la edad.

2.2.2. Sujetos A+N+ versus A+N-

Debido a la poca cantidad de información, este estudio solo pudo evaluar la función cognitiva global y la memoria. Los resultados indican que la diferencia en las puntuaciones entre los A+N+ y A+N- es significativa para la memoria, siendo más bajas para estos últimos.

2.2.3. Resultados

Teniendo en cuenta los resultados anteriores, se deduce que los indicadores neuropsicológicos son sensibles a las diferentes fases del AD preclínico en sujetos CU. Esto implica que la detección de la enfermedad se puede realizar de forma aun más temprana de lo que se creía, siendo los marcadores neuropsicológicos potenciales indicadores. Por lo tanto, se propone la categoría C+ que denota alteraciones cognitivas, precursoras de cambios en los biomarcadores. Finalmente, se indica la necesidad de determinar puntos de corte con el fin de definir rangos dentro de lo que se considera cognitivamente normal. Para ello, se hacen varias sugerencias como usar conjuntos de biomarcadores, datos longitudinales, o considerar la queja subjetiva de memoria, puesto que es el dominio que mejor caracteriza al AD.

2.3. Estimating long-term multivariate progression from short-term data

2.3.1. Introducción

Frecuentemente se usan modelos de efectos mixtos para analizar la progresión de una enfermedad a lo largo del tiempo. Las enfermedades con largos períodos de evolución se

suelen analizar a corto plazo en diferentes etapas para tener una imagen del largo plazo. Sin embargo, es necesario tener una referencia, que en el caso de la AD (p.ej. inicio de la demencia) puede resultar imprecisa. Esto se traduce en dificultades a la hora de situar los resultados en el dominio del tiempo. El autor propone un método SEMOR (regresión auto modelada) con efectos lineales simples a nivel de sujeto, mientras que se usará un suavizado monótono no paramétrico en el largo plazo.

Los datos usados en el algoritmo provienen de ADNI (figura 2.3). Debido a que el subgrupo de sujetos que desarrollan AD es reducido, la información se complementa con la de PAQUID. Aunque esta última no incluye datos de biomarcadores recientes, puede servir para ajustar las curvas a largo plazo.

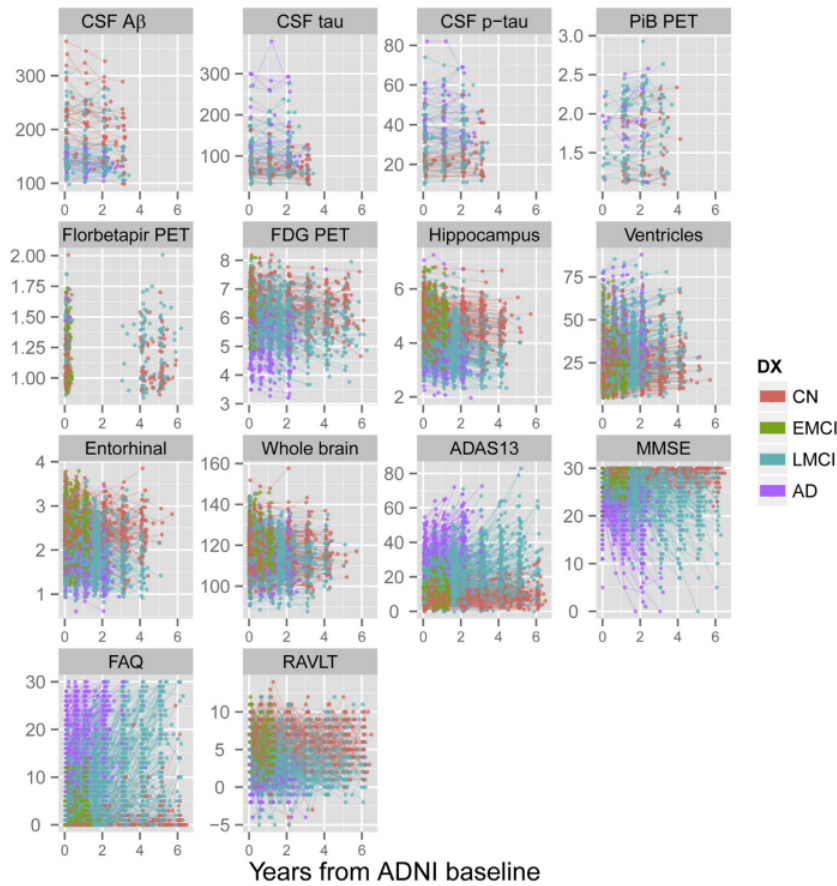


Figura 2.3: Panel de biomarcadores de ADNI. Fuente: Donohue et al.[28]

2.3.2. Modelo y algoritmo

La ecuación siguiente describe el modelo empleado:

$$Y_{ij} = g_j(t + \gamma_i) + \alpha_{0ij} + \alpha_{1ij}t + \epsilon_{ij}(t)$$

Donde:

- i es al individuo.
- j es la medida.

- t es el tiempo de observación a corto plazo
- $t + \gamma_i$ es la progresión a largo plazo. Por lo tanto, γ_i corresponde al desplazamiento temporal (desconocido). Tiene media cero y varianza σ_γ^2 . Este desplazamiento temporal es específico para cada sujeto, ya que la enfermedad se manifiesta a diferentes edades.
- g_j es una función continua monótona diferenciable.
- $(\alpha_{0ij}, \alpha_{1ij})$ son bivariantes gaussianas de media cero y matriz de covarianza Σ_j .
- $\epsilon_{ij}(t)$ son errores residuales gaussianos independientes de media cero varianza σ_j .

El algoritmo reduce un problema de elevada complejidad a problemas más sencillos. Los parámetros desconocidos (g_j , γ_i , α) se estiman usando el resto de parámetros. De esta forma, se itera hasta la convergencia de la RSS (suma de cuadrados residuales). Se usan los residuos parciales $R_{ij}^g(t)$, $R_{ij}^\alpha(t)$, $R_{ij}^\gamma(t)$.

El algoritmo comienza inicializando $\gamma_i = 0$ e iterando sobre los siguientes pasos:

1. Dado γ_i , estimar g_j considerando $\alpha_{0ij} = \alpha_{1ij} = 0$ e iterando sobre la siguiente subrutina:
 - a) Estimar g_j mediante un suavizado monótono de $R_{ij}^g(t)$.
 - b) Estimar α_{0ij} , α_{1ij} por el modelo lineal mixto de $R_{ij}^\alpha(t)$. Repetir los pasos a y b hasta la convergencia de la RSS para la jth medida:

$$RSS_j = \sum_{it} [Y_{ij}(t) - g_j(t + \gamma_i) - \alpha_{0ij} - \alpha_{1ij}t]^2$$

2. Dado el conjunto de curvas g_j , fijar $\alpha_{0ij} = \alpha_{1ij} = \epsilon_{ij}(t) = 0$, y estimar cada γ_i con la media de $R_{ij}^\gamma(t)$ para todo j y t . Repetir pasos 1 y 2 hasta que la convergencia total RSS sea $\sum_{it} [Y_{ij}(t) - g_j(t + \gamma_i) - \alpha_{0ij} - \alpha_{1ij}t]^2$.

2.3.3. Resultados

Antes de enviar los datos de ADNI al algoritmo, fueron transformados a una escala de percentiles (figura 2.4). La escala abarca de 0 (valor más leve observado) a 100 (valor más grave observado). Los percentiles se obtuvieron usando una función de distribución acumulativa empírica, en la que se ponderó de acuerdo a las observaciones en cada categoría (CN, EMCI, LMCI, AD).

El algoritmo se aplicó a un grupo positivo en $A\beta$ y en $APOE \epsilon 4$ (figura 2.5), ya que podrían ser precursores de la AD. El enfoque no paramétrico es más flexible que el empleado por Jack et al, en el que se asumen sigmoides. El CSF $A\beta$ evoluciona de forma lineal; mientras que tau, pTau y PiB PET siguen sigmoides. El metabolismo de glucosa, volumen del hipocampo, volumen ventricular y ADAS13 siguen una trayectoria lineal. En contraste, Function sigue una trayectoria parabólica.

Como conclusión se tiene que PiB PET es el primero en alcanzar niveles anormales. Le siguen CSF tau y pTau. El aprendizaje, metabolismo de glucosa, atrofia del hipocampo y la cognición son las próximas en la sucesión. Function es el último marcador en progresar a un estado anormal. Además, el volumen del hipocampo mediante MRI es el marcador que aporta resultados más eficientes en todo el transcurso de la enfermedad.

Outcome	n/N*	Percentiles				
		0	25 th	50 th	75 th	100 th
CSF tau, pg/mL	106/406	31	64	90	122	379
CSF p-tau, pg/mL	106/406	10.0	21.6	32.3	42.0	82.0
CSF amyloid- β , pg/mL	106/406	364	181	146	131	98
PiB PET, SUVR	103/224	1.09	1.36	1.85	2.09	2.93
Florbetapir PET, SUVR	910/910	0.83	1.01	1.21	1.41	2.01
FDG PET uptake	1261/2806	8.54	6.56	6.06	5.50	3.20
Ventricles, % ICV	1526/4864	0.45	1.67	2.44	3.40	9.03
Hippocampus, % ICV	1526/4864	0.79	0.50	0.43	0.36	0.20
ADAS13	1592/6314	0.00	9.23	15.91	26.26	85.00
MMSE	2396/7216	30.0	29.2	27.5	24.4	0.0
FAQ	1590/6402	0.00	0.73	4.04	12.87	30.00
RAVLT (Trial 5-Trial 1)	1597/6341	14.00	6.37	4.08	2.18	-5.00

Figura 2.4: Tabla de percentiles de ADNI. Siendo n y N el número total de sujetos y observaciones, respectivamente. Fuente: Donohue et al.[28]

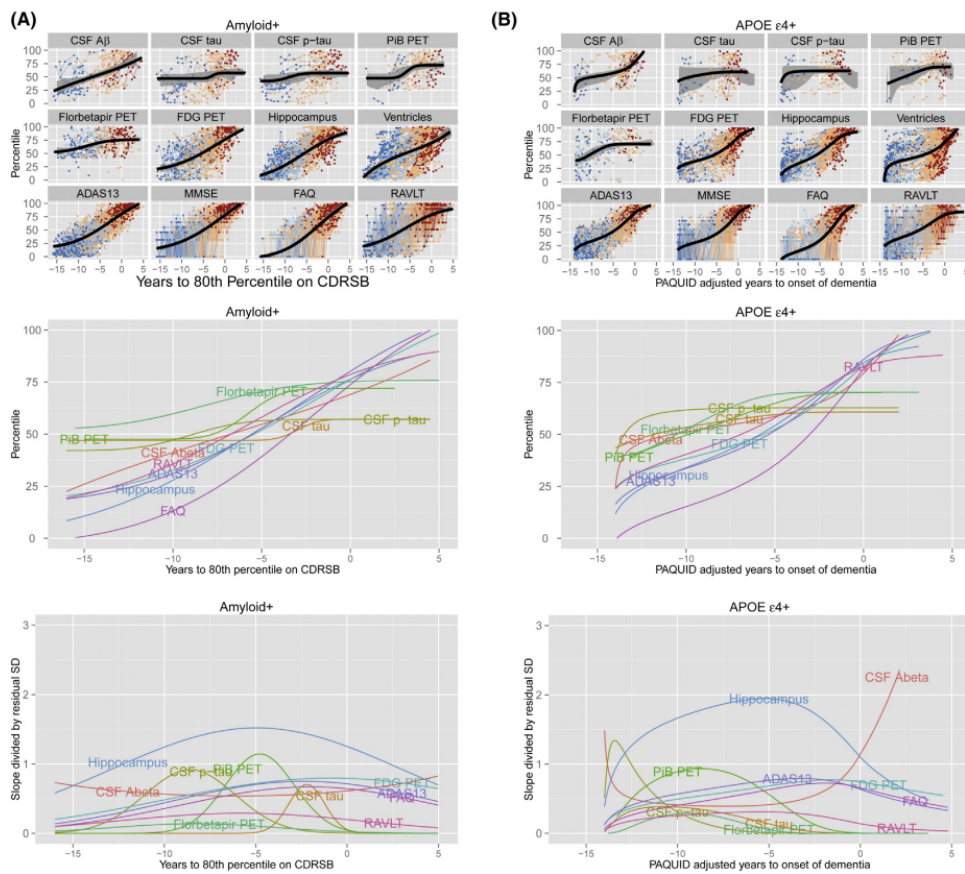


Figura 2.5: Trayectorias a largo plazo de sujetos con niveles anormales de $A\beta$ (A) y de $APOE \epsilon 4$ (B). Fuente: Donohue et al.[28]

2.4. Learning Multimodal Digital Models of Disease Progression from Longitudinal Data: Methods & Algorithms for the Description, Prediction and Simulation of Alzheimer's Disease Progression

La geometría Riemanniana proporciona un marco de trabajo adecuado para estudiar la progresión de enfermedades, tal como la enfermedad de Alzheimer. A modo de ejemplo, se podría describir el crecimiento de un niño tomando fotos en a distintas edades. Claramente, la imagen resultante entre 2 instantes temporales no es la media entre los valores de los píxeles en dichos instantes. La figura 2.6 refleja esta idea intuitivamente. La curva negra representa la trayectoria de un individuo. La superficie azul y roja es el subespacio resultante de todas las posibles trayectorias de los individuos.

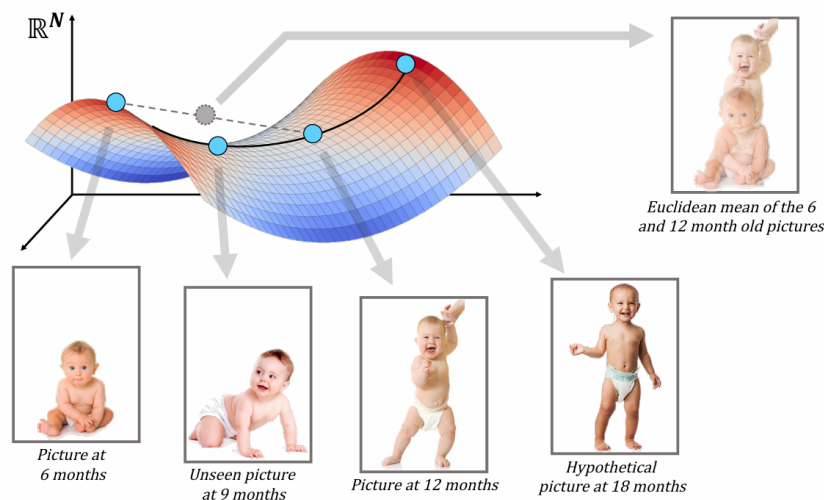


Figura 2.6: Subespacio (superficie azul y roja) del espacio \mathbb{R}^N correspondiente a todas las posibles imágenes del crecimiento de un niño. Las fotos son conocidas a los 6 meses y 12 meses. La media Euclídea entre ambas resulta en una superposición borrosa de las mismas. La curva negra representa la trayectoria de un sujeto, permitiendo estimar las imágenes a los 9 meses y 18 meses. Fuente: tesis de Igor Koval. [38]

2.4.1. Geometría de Riemann

Una variedad es un espacio topológico en el que cada punto es localmente homeomorfo al espacio Euclídeo, es decir, se puede tratar a nivel local como espacio Euclídeo. En las variedades suaves (no hay saltos entre estructuras locales), cada punto tiene asociado un plano tangente en sus vecindades. Dicha variedad será Riemanniana si se le asocia una colección de productos internos $g_{\mathbf{M}} = (g_{\mathbf{p}})_{\mathbf{p} \in \mathbf{M}}$. A cada punto \mathbf{p} le corresponde el producto interno $g_{\mathbf{p}}$ sobre el campo vectorial del plano tangente $\mathbf{T}_{\mathbf{p}}\mathbf{M}$.

Las geodésicas son el equivalente en la geometría Riemanniana a las líneas rectas en el espacio euclídeo: representan el camino más corto entre 2 puntos de la variedad. Una curva suave $\gamma : \mathbf{I} \subset \mathbb{R} \rightarrow \mathbf{M}$ es geodésica en \mathbf{M} si $\nabla_{\dot{\gamma}} \dot{\gamma} = 0$ (siendo ∇ la conexión de Levi-Civita).

Otros conceptos importantes son los de mapeo exponencial y transporte paralelo. El primero representa un paso en la variedad, desde el punto \mathbf{p} en la dirección \mathbf{v} hasta el punto alcanzado en el instante siguiente, $t + 1$. Formalmente: $\mathbf{v} \in \mathbf{T}_{\mathbf{p}}\mathbf{M} \rightarrow \text{Exp}_{\mathbf{p}}(\mathbf{v}) = \text{Exp}_{\mathbf{p},t}(\mathbf{v})(t + 1)$. El segundo consiste en trasladar un vector a lo largo de una curva suave

sin afectar a su orientación, es decir, que permanezca paralelo.

2.4.2. Modelos de progresión de la enfermedad

La progresión de la enfermedad de Alzheimer puede caracterizarse por un modelo de efectos mixtos. Los efectos fijos corresponden a la trayectoria general de la población, y los efectos aleatorios representan las trayectorias individuales. Cada observación, y_{ij} pertenece a una variedad Riemanniana $\mathbf{M} \subset \mathbb{R}^N$. La trayectoria general de la población viene dada por la geodésica $\gamma_0 : \mathbf{I} \subset \mathbb{R} \rightarrow \mathbb{R}^N$ que pasa por el punto \mathbf{p}_0 con una velocidad \mathbf{v}_0 en el instante t_0 . Las trayectorias individuales son variaciones espacio-temporales de la trayectoria general:

- Variación espacial: la trayectoria general se traslada en la dirección de $\mathbf{w}_i \in \mathbf{T}_{\mathbf{p}_0}\mathbf{M}$. Esto se logra por mapeo exponencial de los vectores $P_{\gamma_0, t_0, t}(\mathbf{w}_i)$, obtenidos por transporte paralelo de \mathbf{w}_i a lo largo de la curva γ_0 . Formalmente, la trayectoria individual obtenida se expresa:

$$\eta^{\mathbf{w}_i}(t) := \text{Exp}_{\gamma_0(t)}(P_{\gamma_0, t_0, t}(\mathbf{w}_i))$$

La figura 2.7 refleja esta idea.

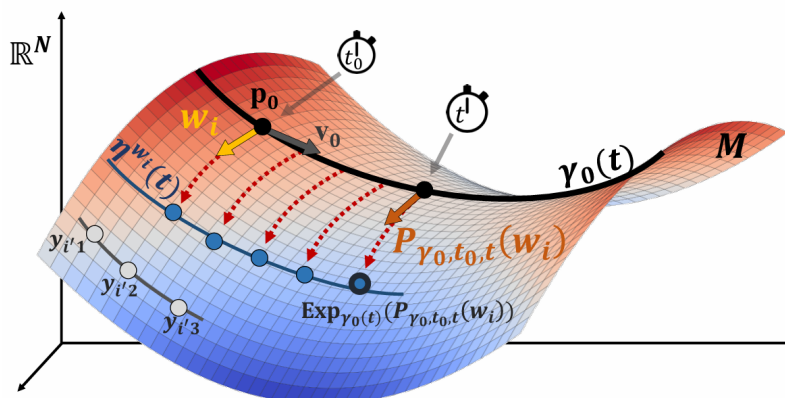


Figura 2.7: Descripción geométrica del modelo donde cada observación $y_{ij} \in \mathbb{R}^N$ pertenece a la variedad Riemanniana \mathbf{M} . La trayectoria general de la población viene dada por la geodésica $\gamma_0(t)$. La trayectoria individual $\eta^{\mathbf{w}_i}(t)$ corresponde al mapeo exponencial de los vectores $P_{\gamma_0, t_0, t}(\mathbf{w}_i)$ obtenidos a partir de $\mathbf{w}_i \in \mathbf{T}_{\mathbf{p}_0}\mathbf{M}$ por transporte paralelo a lo largo de $\gamma_0(t)$. Fuente: tesis de Igor Koval. [38]

- Variación temporal: se consideran 2 efectos:
 - Factor de aceleración (α_i): representa la velocidad con la que progresa cada sujeto. Si $\alpha_i > 1$ (resp. $\alpha_i < 1$), el sujeto progresa más rápido (resp. más lento) que $\gamma_0(t)$.
 - Traslación temporal (τ_i): representa el inicio de la enfermedad respecto de γ_0 . Si $\tau_i > 0$ (resp. $\tau_i < 0$), el sujeto empezará la enfermedad más tarde (resp. antes) que la población general.

La combinación de ambos efectos permite llevar a cabo una reparametrización temporal con el fin de alinear las trayectorias (i.e. busca una escala donde cualquier instante de tiempo represente el mismo evento para todos los sujetos):

$$\psi_i : t \rightarrow \alpha_i(t - \tau_i - t_0) + t_0$$

Con el fin de reducir la dimensionalidad del espacio de características, se descompone \mathbf{w}_i por medio de un Análisis de Componentes Independientes (ICA), de forma que:

$$\mathbf{w}_i = A\mathbf{s}_i$$

Donde $A \in \mathbb{R}^{N \times N_s}$ es la matriz de mezcla, N es la dimensión inicial del espacio de características, N_s es la dimensión del espacio de características después de aplicar el ICA, y $\mathbf{s}_i = (s_{ij})_{1 \leq j \leq N_s}$ son las fuentes.

Adicionalmente, conviene remarcar que el vector \mathbf{w}_i debe ser ortogonal a la velocidad \mathbf{v}_0 con el fin de no interferir con la progresión temporal. Esto se logra imponiendo la condición $g^M(\mathbf{w}_i, \mathbf{v}_0) = 0$ (o $g^M(A_k, \mathbf{v}_0) = 0 \quad \forall k \in \{1, \dots, N\}$ si se aplica ICA, siendo A_k la k -ésima componente de la matriz de mezcla A).

La variedad considerada se compone del producto de variedades unidimensionales tal que $\gamma_0(t) = (\gamma_1(t), \dots, \gamma_n(t))$. Las curvas propuestas y sus ecuaciones se presentan en la tabla 2.1.

2.4.3. Métodos

Sea el modelo del sujeto i -ésimo:

$$\mathbf{y}_i = f(\theta, \mathbf{z}_i) + \epsilon_i$$

Donde:

- \mathbf{y}_i son las variables observadas.
- \mathbf{z}_i son las variables aleatorias correspondientes al sujeto i -ésimo.
- f es el modelo de la enfermedad.
- θ son los parámetros del modelo, compuestos por:
 - Parámetros de la función f .
 - Parámetros asociados a las distribuciones de las variables aleatorias.

El objetivo es encontrar los parámetros que maximicen la verosimilitud:

$$p(\mathbf{y}; \theta) = \int p(\mathbf{y}, \mathbf{z}; \theta) d\mathbf{z} = \int p(\mathbf{y}|\mathbf{z}; \theta)p(\mathbf{z}; \theta) d\mathbf{z}$$

Dado que encontrar los valores de dichos parámetros puede no ser directo, se usan los siguientes algoritmos:

- **Algoritmo E-M** (*Expectation-Maximization*)
- **Algoritmo SAEM** (*Stochastic Approximation Expectation Maximization*)
- **Algoritmo MCMC-SAEM** (*Monte Carlo Markov Chain MCMC-SAEM*)
- **Algoritmo Hasting Metropolis within Gibbs sampler**

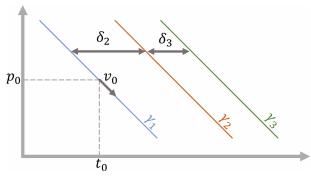
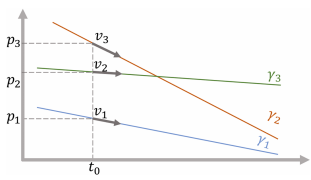
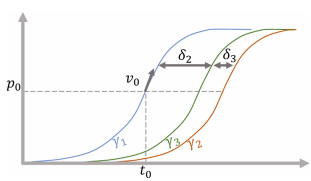
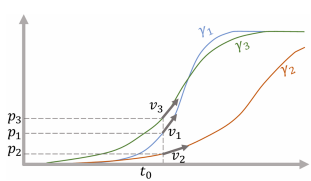
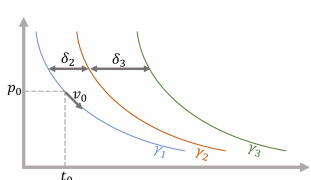
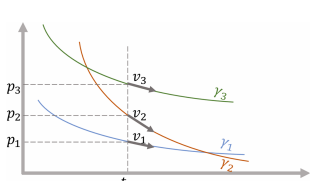
Curvas	Ecuaciones
 <p>(a) Rectas paralelas</p>	$g_p^k(u, v) = uv$ $\gamma_k(t) = p_0 + (t - t_0 - \delta_k)v_0$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = w_{ik} + p_0 + (\alpha_i(t_{ij} - t_0 - \tau_i) + \delta_k)v_0$
 <p>(b) Rectas</p>	$g_p^k(u, v) = uv$ $\gamma_k(t) = p_k + (t - t_0)v_k$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = w_{ik} + p_k + \alpha_i(t_{ij} - t_0 - \tau_i)v_k$
 <p>(c) Logísticas paralelas</p>	$g_p^k(u, v) = \frac{uv}{p^2(1-p)^2}$ $\gamma_k(t) = \left(1 + \left(\frac{1}{p_0} - 1\right)E(t)\right)^{-1}$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = \left(1 + \left(\frac{1}{p_0} - 1\right)\exp\left(\frac{-v_0(\alpha_i(t - \tau_i - t_0) + \delta_k)}{p_0(1-p_0)} - \frac{w_{ik}(1 + gE(t_0))^2}{gE(t_0)}\right)\right)^{-1}$
 <p>(d) Logísticas</p>	$g_p^k(u, v) = \frac{uv}{p^2(1-p)^2}$ $\gamma_k(t) = \left(1 + \left(\frac{1}{p_k} - 1\right)\exp\left(-\frac{v_k}{p_k(1-p_k)}(t - t_0)\right)\right)^{-1}$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = \left(1 + \left(\frac{1}{p_k} - 1\right)\exp\left(-\frac{w_{ik} + v_k\alpha_i(t_{ij} - t_0 - \tau_i)}{p_k(1-p_k)}\right)\right)^{-1}$
 <p>(e) Exponenciales paralelas</p>	$g_p^k(u, v) = \frac{uv}{p^2}$ $\gamma_k(t) = p_0 \exp\left(-\frac{v_0}{p_0}(t - t_0 + \delta_k)\right)$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = p_0 \exp\left(-\frac{w_{ik}}{p_0} - \frac{v_0}{p_0}(\alpha_i(t_{ij} - t_0 - \tau_i) + \delta_k)\right)$
 <p>(f) Exponenciales</p>	$g_p^k(u, v) = \frac{uv}{p^2}$ $\gamma_k(t) = p_k \exp\left(-\frac{v_k}{p_k}(t - t_0)\right)$ $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij})) = p_k \exp\left(-\frac{w_{ik}}{p_k} - \frac{v_k\alpha_i}{p_k}(t_{ij} - t_0 - \tau_i)\right)$

Tabla 2.1: Métrica $g_p^k(u, v)$, ecuación de la geodésica $\gamma_k(t)$ y de la trayectoria individual $\eta_k^{\mathbf{w}^i}(\psi_i(t_{ij}))$ para distintos tipos de evoluciones temporales propuestas: rectas paralelas (a), rectas (b), logísticas paralelas (c), logísticas (d), exponenciales paralelas (e), exponenciales (f). Se considera la progresión de 3 variables. Para mayor claridad, se escribe $E(t) = \exp\left(\frac{-v_0}{p_0(1-p_0)}(t + \delta_k - t_0)\right)$. Fuente: tesis de Igor Koval. [38]

Estimación de los modelos de progresión de la enfermedad

Particularizando a los modelos de progresión de la enfermedad, se tiene:

$$y_{ij} = f(\theta_{geom}, \mathbf{z}_i, t_{ij}) + \epsilon_{ij}$$

Donde:

- $\theta_{geom} = ((p_k)_{1 \leq k \leq N}, (v_k)_{1 \leq k \leq N}, t_0, (\beta_k)_{1 \leq k \leq (N-1)N_s})$ son los efectos fijos. Representan la progresión general de la población. Dependen del tipo de curva propuesta.
- $\mathbf{z}_i = (\alpha_i, \tau_i, (s_{ij})_{1 \leq j \leq N_s})$ son los efectos aleatorios. Representan las variaciones individuales de la progresión general de la población.
- $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$

A continuación, es posible aplicar los siguientes procesos estadísticos:

- **Calibración:** consiste en estimar los parámetros $\hat{\theta}$ que mejor describen la trayectoria general de la población. El algoritmo MCMC-SAEM resulta de interés.
- **Personalización:** conocido $\hat{\theta}$, se estiman las variables \mathbf{z}_i^* que mejor se ajustan al sujeto. Se utiliza el método L-BFGS o el método de Powell. Otra alternativa consiste en ejecutar K iteraciones del algoritmo MCMC-SAEM únicamente con los parámetros individuales del sujeto.
- **Reconstrucción, imputación de valores faltantes y predicción:** dados $\hat{\theta}$ y \mathbf{z}_i^* , se estima $\tilde{y}_{ij} = f(\theta_{geom}, \mathbf{z}_i^*, t_{ij})$, distinguiéndose entre:
 - **Reconstrucción:** t_{ij} ha sido observado ($t_{ij} \in \{t_{i1}, t_{ik_i}\}$).
 - **Imputación de valores faltantes:** t_{ij} se encuentra entre la primera y la última visita ($t_{i1} < t_{ij} < t_{ik_i}$).
 - **Predicción:** t_{ij} corresponde a un instante de tiempo futuro ($t_{ij} > t_{ik_i}$).
- **Simulación:** dados $\hat{\theta}$ y un conjunto de variables individuales \mathbf{z}_i , se crean nuevas variables individuales \mathbf{z}_i^v que representan nuevos sujetos virtuales. La calidad de los sujetos simulados se puede medir comparando las distribuciones de los datos virtuales y originales. Se puede usar el método KDE (*Kernel Density Estimation*).

2.5. Robust parametric modeling of Alzheimer's disease progression

Los modelos paramétricos son aquellos en los que los datos se describen mediante un conjunto finito de parámetros. Son simples, rápidos y requieren de menos datos en el entrenamiento. En contrapartida, es necesario el alineamiento temporal de las trayectorias de los sujetos. Esto implica transformar la edad del sujeto a una escala común, DPS (*Disease Progression Score*), sobre la que se representa la evolución de la enfermedad.

El modelo propuesto utiliza la estimación M como método de regresión robusta, en la que el impacto de los valores atípicos en la función de pérdidas se atenúa.

2.5.1. Métodos

Los estimadores M son una generalización de la regresión por mínimos cuadrados y la estimación de máxima verosimilitud. Sea $y_{i,j,k}$ la medida de un biomarcador k , para una visita j y un sujeto i :

$$y_{i,j,k} = f(s_{i,j}; \theta_k) + \sigma_k \epsilon_{i,j,k}$$

Donde $f(s_{i,j}; \theta_k)$ es una función con forma de S, θ_k son los parámetros del modelo, σ_k es la desviación típica del marcador k , $\epsilon_{i,j,k}$ es ruido blanco aditivo Gaussiano, y $s_{i,j}$ es el DPS del sujeto i para la visita j , definido como:

$$s_{i,j} = \alpha_i t_{i,j} + \beta_i$$

Donde $\alpha_i \in \mathbb{R}_{>0}$ y $\beta_i \in \mathbb{R}$ son la velocidad de progresión y el tiempo de inicio de la enfermedad para cada sujeto, respectivamente. Estos parámetros son los que permiten transformar la edad de cada sujeto individual a la escala temporal de la enfermedad, DPS (figura 2.8).

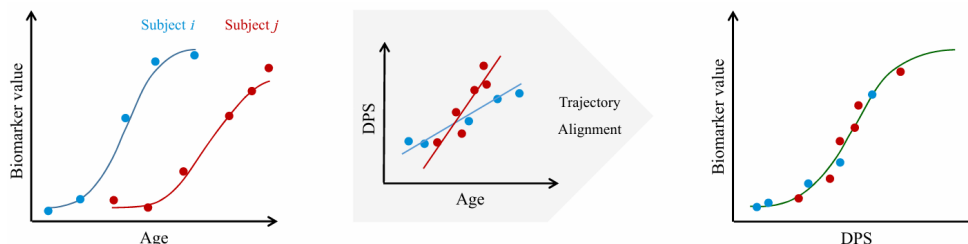


Figura 2.8: Transformación de la edad a una escala común, DPS. Izquierda: función ajustada a los marcadores de cada sujeto. Medio: alineación de las trayectorias, transformando la edad a DPS. Derecha: función ajustada a la trayectoria alineada de todos los sujetos. Fuente: Ghazi et al.[31]

Se trata de un problema de optimización multiobjetivo:

$$\{\hat{\alpha}, \hat{\beta}, \hat{\theta}\} = \min_{i,j,k} \sum w_i \rho \left(\frac{y_{i,j,k} - f(s_{i,j}; \theta_k)}{\sigma_k} \right)$$

Donde $\rho(\cdot)$ es una función de máxima verosimilitud y $w_i = 1/N_i$, es un factor de ponderación que equilibra la influencia de cada sujeto según su número de visitas, N_i . Cabe decir que en el caso de $\rho = (\cdot)^2$ y $f(s_{i,j}; \theta_k) = \theta_{k,1}s_{i,j} + \theta_{k,0}$ se trataría de una regresión lineal por mínimos cuadrados.

Las funciones consideradas que ajustan la trayectoria de los marcadores se presentan en la tabla 2.2. Todas tienen rango (0,1) y presentan un punto de inflexión en $c \in \mathbb{R}$. Esto último permite conocer el orden en el que empiezan a cambiar los marcadores según evoluciona la enfermedad. Los parámetros $b \in \mathbb{R}_{>0}$ y $\gamma \in \mathbb{R}_{>0}$, correspondientes con la velocidad de crecimiento y la simetría respectivamente, permiten modelar patrones asimétricos.

Función	$g(s; \theta)$
Verhulst	$[1 + e^{-b(s-c)}]^{-1}$
Gompertz	$e^{-e^{-b(s-c)}}$
Richards	$[1 + \gamma e^{(-b(s-c))}]^{-1-\gamma}$
Stannard modificada	$[1 + \frac{1}{\gamma} e^{-\frac{b}{\gamma}(s-c)}]^{-\gamma}$

Tabla 2.2: Funciones logísticas consideradas para ajustar las trayectorias de los marcadores. Fuente: Ghazi et al.[31]

Los valores de las asíntotas horizontales se ajustan mediante los parámetros a y d :

$$f(s; \theta) = (a - d)g(s; \theta) + d$$

Conviene fijar estos parámetros en el caso de que la escala de los marcadores esté restringida a un determinado rango de parámetros.

El algoritmo usado se describe a continuación:

1. Inicialización de $\{\alpha^{(0)}, \beta^{(0)}, \theta^{(0)}\}$.
2. Iterar l veces hasta la convergencia:
 - a) Estimar los parámetros de los marcadores (θ) manteniendo fijos los parámetros de los sujetos (α y β):

$$\hat{\theta}^{(l)} = \min_{i,j,k} \sum_{i,j,k} w_i \rho \left(\frac{y_{i,j,k} - f(\hat{\alpha}_i^{l-1} t_{i,j} + \hat{\beta}_i^{l-1}; \theta_k)}{\sigma_k} \right)$$

- b) Estimar los parámetros de los sujetos manteniendo fijos los parámetros de los marcadores:

$$\{\hat{\alpha}^{(l)}, \hat{\beta}^{(l)}\} = \min_{i,j,k} \sum_{i,j,k} w_i \rho \left(\frac{y_{i,j,k} - f(\alpha_i t_{i,j} + \beta_i; \hat{\theta}_k^{(l)})}{\sigma_k} \right)$$

3. Estandarización del DPS respecto de las visitas cognitivamente normales:

$$\begin{aligned} s_{i,j} &= (s_{i,j} - \mu_{cn}) / \sigma_{cn} \\ \alpha_i &= \alpha_i / \sigma_{cn} \\ \beta_i &= (\beta_i - \mu_{cn}) / \sigma_{cn} \\ b_k &= \sigma_{cn} b_k \\ c_k &= (c_k - \mu_{cn}) / \sigma_{cn} \end{aligned}$$

De esta forma, se tiene una mejor intuición del punto en el que se encuentra cada sujeto. Los sujetos cognitivamente normales presentan un DPS centrado en 0 con una variabilidad de 1. Por encima de un DPS igual a 1, el sujeto estará en deterioro cognitivo.

Los estimadores M descritos en la tabla 2.3 buscan reducir la influencia de los valores atípicos.

Función de pérdidas	$\rho(r)$
L2	r^2
L1-L2	$2(\sqrt{1+r^2} - 1)$
Logística	$\ln(\cosh(r))$
Huber modificada	$\begin{cases} 1 - \cos(r), & r \leq \pi/2 \\ r + (1 - \pi/2), & r > \pi/2 \end{cases}$
Cauchy-Lorentz	$\ln(1 + r^2)$

Tabla 2.3: Estimadores M considerados para la regresión robusta. Fuente: Ghazi et al.[31]

Finalmente, se utiliza el método KDE (*Kernel Density Estimation*) para ajustar la función de densidad del DPS. Con un clasificador de Bayes se estima la clase a la que pertenece cada sujeto.

Datos

Las bases de datos utilizadas en el estudio son ADNI y NACC (*National Alzheimer's Coordinating Center*). Los datos se particionan en 2 conjuntos *train* y *test* equilibrados según los diagnósticos inicial y final y el número de visitas de cada sujeto. Además, se aplica un proceso de *bootstrapping* con reemplazamiento siguiendo el mismo criterio para la selección de los sujetos.

Medidas de calidad

Las medidas de calidad evaluadas son:

- BIC (*Robust Bayesian information criterion*) en la selección de los modelos:

$$\text{BIC} = 2E_{train}^{(L_{opt})} + Q\ln(N)$$

Donde $2E_{train}^{(L_{opt})}$ es la pérdida en la iteración óptima L_{opt} , N es el número de medidas, y Q es el número total de parámetros. De esta forma, los modelos con mayor número de parámetros quedan penalizados.

- MAE (*Mean Absolute Error*) en la evaluación del rendimiento.
- AUC (*Area Under the Curve*) en la evaluación del diagnóstico.

Criterio de parada

El criterio de parada se rige por:

$$\{\hat{\alpha}, \hat{\beta}, \hat{\theta}\} = \min_{L_{min} \leq l \leq L_{max}} E_{valid}^{(l)}$$

Donde $E_{valid}^{(l)}$ es la pérdida en el *set* de validación en la iteración l , L_{min} es el número mínimo de iteraciones, y L_{max} el número máximo de iteraciones.

2.5.2. Resultados

Algunos de los resultados obtenidos en el estudio son:

- La función de Stannard con la pérdida logística logran el menor BIC entre las combinaciones consideradas.
- Los marcadores de CSF, PET y RAVLT-IR son los primeros en evolucionar. Los *test* RAVLT detectan anomalías de forma temprana, mientras que las pruebas cognitivas y de MRI lo hacen más tarde (figura 2.9).
- Los resultados obtenidos son generalizables a distintas cohortes sin pérdidas significativas.

2.6. Temporal modeling and AT profiles in the early phase of Alzheimer's disease

La enfermedad de Alzheimer se caracteriza por su biología subyacente. Inicialmente, se manifiesta a partir de placas $A\beta$, seguidas por ovillos neocorticales de tau, incluso en

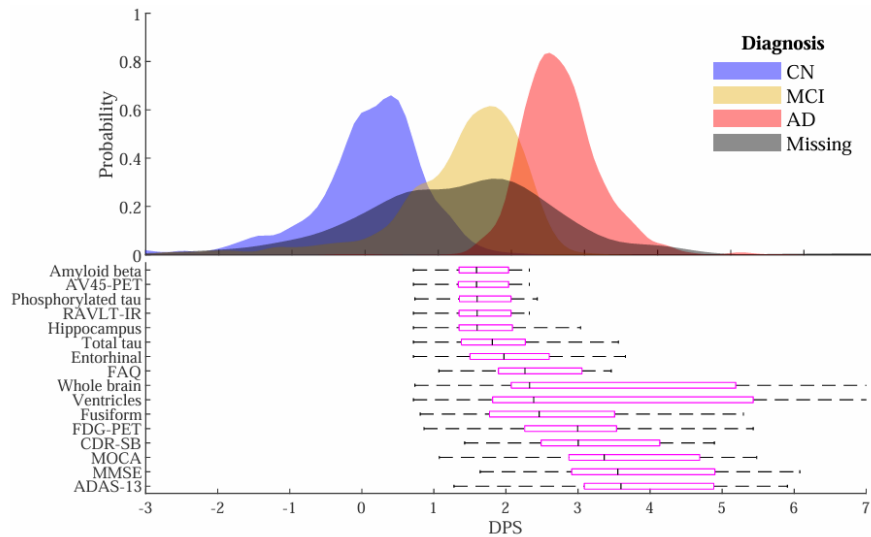


Figura 2.9: Probabilidad estimada del DPS según el estado de los sujetos (CN, MCI, AD). Los diagramas de caja indican los percentiles 25 y 75 de los puntos de inflexión estimados para cada biomarcador, marcando en la caja la mediana. Los bigotes representan los puntos de inflexión no anómalos más extremos. Fuente: Ghazi et al.[31]

pacientes asintomáticos. Los síntomas clínicos pueden ser causados por otros trastornos, siendo insuficientes para diagnosticar AD. Los sujetos pueden clasificarse según su perfil AT, donde A+ representa niveles anormales de amiloide y T+ niveles anormales de tau. El objetivo de este estudio consiste en integrar el perfil AT con DPM que reconstruyen la historia natural de la enfermedad.

2.6.1. Métodos

Los sujetos analizados forman parte de la base de datos de ADNI, de los cuales se requiere al menos un escáner T1-MRI. También son requeridos datos CSF para determinar los perfiles AT. Los criterios para diagnosticar patología amiloide (A+) son $p\text{Tau}/A\beta_{42} > 0,028$ o $A\beta_{42} < 880$ pg/mL, y $p\text{Tau} > 27$ pg/mL para patología tau (T+). Los sujetos se clasifican en sCU (rápidos si tienen seguimiento por menos de 4 años y lentos en caso contrario) y pCU (rápidos si convierten en menos de 4 años y lentos en caso contrario).

El procedimiento seguido (figura 2.10) consta de dos etapas:

1. Se modelan las trayectorias longitudinales de los marcadores mediante LME y se construye un modelo de Cox extendido en cada instante temporal. Los ratios de riesgo se transforman a través de una regresión logística en probabilidades de conversión de CU a MCI. La selección y evaluación de características se realiza mediante validación cruzada anidada:
 - a) El bucle interno utiliza el algoritmo de mínima redundancia máxima relevancia (mRMR) para proponer conjuntos de características capaces de predecir la progresión de CU a MCI. Con el objetivo de determinar los mejores conjuntos, se utiliza un método de remuestreo basado en diferencias de información mutua en mRMR.
 - b) El bucle externo se encarga de evaluar los conjuntos de características propuestos en base a la sensibilidad, especificidad, precisión y área bajo la curva.

2. Los marcadores que mejores puntuaciones consiguen en la evaluación de los modelos predictivos se utilizan para entrenar DPM con GRACE. La evaluación de GRACE sigue 3 criterios:

- Determinación de las siguientes puntuaciones:
 - $SCORE_1 = \frac{\#\{i|(i \in sCU) \cap ((t_{end}^c + \delta_i) < t_{onset})\}}{\#sCU}$
 - $SCORE_2 = \frac{\#\{i|(i \in pCU) \cap ((t_{i_1}^c + \delta_i) < t_{onset})\}}{\#pCU}$
 - $SCORE_3 = \frac{\#\{i|(i \in pCU) \cap ((t_{end}^c + \delta_i) > t_{onset})\}}{\#pCU}$
- Correlación entre los tiempos de conversión estimados y reales en sujetos pCU.
- Comparación de los valores de los marcadores en las trayectorias a largo plazo en t_{onset} con los valores de corte utilizados en la práctica clínica para diagnosticar MCI, donde:

$$\hat{t}_{onset} = \arg \min_{t_{onset}} \sum_{i=1}^2 \sum_{j=i+1}^3 |SCORE_i(t_{onset}) - SCORE_j(t_{onset})|$$

Además, se utiliza la técnica de *bootstrapping* para obtener intervalos de confianza.

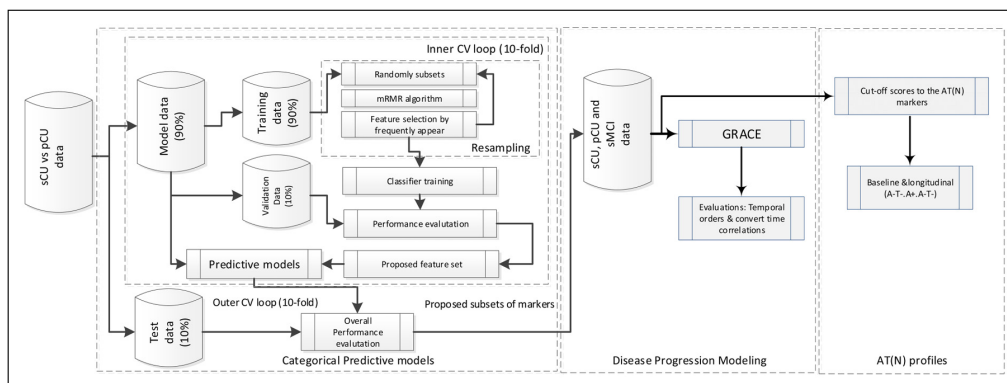


Figura 2.10: Diagrama de flujo que proporciona una visión general de la interacción entre los tres componentes principales que intervienen en el estudio: (a) modelos predictivos categóricos, (b) modelos de progresión de la enfermedad, (c) perfiles AT. Fuente: C. Platero [47].

2.6.2. Resultados

Algunos resultados del estudio se exponen a continuación:

- El criterio $pTau/A\beta_{42} > 0,028$ es más discriminativo que $A\beta_{42} < 880pg/ml$ para clasificar entre sCU y pCU. Lo mismo ocurre con las medidas NM (destacan ADAS13, LDELTOTAL, PACC) frente a marcadores de resonancia magnética o punción lumbar.
- Los factores de mayor riesgo son la edad y la prevalencia de patología amiloide.
- Los modelos predictivos mejoran su puntuación con el tiempo de estudio (tabla 2.4). Se propone el vector $\{ADAS13, CDRSB, AGE\}$ para modelar la progresión de CU a MCI. Aunque no consigue las mejores predicciones, es fácil de implementar en la práctica clínica y da lugar a una de las mejores historias naturales de la enfermedad.

- La población pCU rápida es la más anciana y con mayor prevalencia de patología amiloide. Ocurre lo contrario con la población sCU lenta.
- Las predicciones de CU a MCI y su evolución temporal pueden ser logradas con una sola muestra CSF y un seguimiento con medidas NM exclusivamente.
- Los pacientes A+ al principio del estudio convierten más de 5 años antes que los A- (figura 2.11), atendiendo a medidas NM. Además, presentan un tiempo δ_i mayor.
- Los resultados obtenidos mediante distintos DPM son similares (tabla 2.5). A partir de dichos resultados, se tiene que el tiempo cero ocurre un año y medio después de $t_{ADAS13=12}$ y el comienzo de la patología amiloide 6 años después de $t_{ADAS13=12}$.

Data	SEN (%)	SPE (%)	ACC (%)	AUC	Frequency	Optimal feature subsets
bl	75.3 (73.9 76.4)	70.5 (70.0 71.1)	70.8 (70.3 71.3)	0.797 (0.788 0.804)	1391-1078	A, D, F, L D, F, P C, P
m12	84.2 (83.0 85.5)	71.0 (70.3 71.6)	72.2 (71.6 72.8)	0.847 (0.840 0.854)	1246-1052	A, D, F, L D, F, P C, P
m24	83.0 (81.6 84.4)	77.4 (76.7 78.1)	77.6 (76.9 78.2)	0.877 (0.869 0.884)	1326-1009	A, F, L D, F, P C, P

Tabla 2.4: Puntuaciones para predecir conversores rápidos de CU a MCI. Por cada visita (Inicial - bl, Mes 12 - m12, Mes 24 - m24) se ha construido un modelo predictivo mediante el método Cox extendido. Los números entre paréntesis representan los intervalos de confianza al 95 % (salvo la columna de la frecuencia). A=ADAS13; C= CDRSB; D=DIGITSCORE; F=FAQ; L=LDELTOTAL; P=PACC; AUC=Área bajo la curva; ACC=Exactitud; SEN=Sensibilidad; SPE=Especificidad; Frequency=Valores mínimo y máximo que la combinación de características fue evaluada por la validación cruzada. Fuente: C. Platero [47].

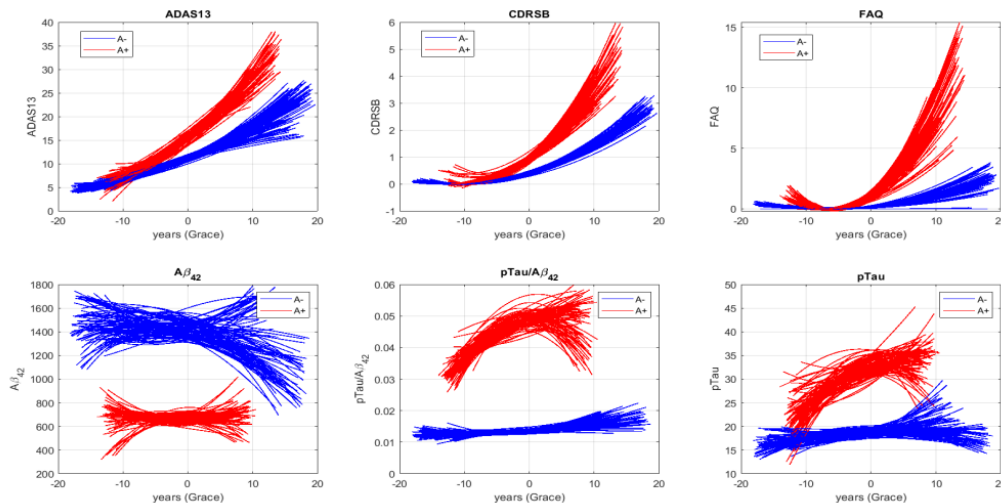


Figura 2.11: Trayectorias a largo plazo de los marcadores en la etapa temprana de AD, comparando sujetos A+ y A- al inicio del estudio. Se ha usado la técnica de bootstrap para obtener intervalos de confianza de las trayectorias. Fuente: C. Platero [47].

Features	tonset	Ecuaciones de regresión	R	SCORE1	SCORE2	SCORE3
A,C,Ag	2.6	$1,08t - 2,83$	0.65	86.0 %	89.2 %	74.2 %
A,C,L,Ag	2.8	$1,09t - 2,95$	0.65	81.5 %	89.4 %	75.3 %
C,L,P,Ag	2.5	$1,24t - 3,52$	0.65	85.5 %	78.8 %	70.6 %
C,F,I,L,M,Ag	2.8	$1,26t - 3,67$	0.64	87.8 %	80.0 %	69.4 %

Tabla 2.5: Vectores de características de los DPM propuestos con sus correlaciones y puntuaciones. A=ADAS13; Ag=Age; C=CDRSB; F=FAQ; I=RAVLT Immediate; L=RAVLT Learning; M=MMSE; P=PACC. Fuente: C. Platero [47].

2.7. Predicting Alzheimer’s disease progression using deep recurrent neural networks

Nguyen et al. proponen el uso de RNN (*Recurrent Neural Networks*) para predecir la evolución de la enfermedad de Alzheimer. A diferencia de los modelos de efectos mixtos, no se asume una trayectoria a priori de los biomarcadores, lo cual puede resultar ventajoso. También permiten trabajar con sujetos con distinto número de visitas y realizar estimaciones en un número indefinido de puntos futuros, en contraste con otros algoritmos como las SVM/SVR. Concretamente, se utiliza una *minimalRNN*, la cual es menos propensa al sobreajuste al tener menos parámetros que las LSTM (*Long Short-Term Memory*). Los resultados se comparan con cuatro modelos de referencia: predicción constante, SVM/SVR, LSS, LSTM.

Además, se exploran tres estrategias para tratar los valores faltantes, un problema recurrente cuando se trabaja con datos longitudinales. Las dos primeras se realizan en la etapa de preprocesamiento. Se imputan los valores faltantes con el último valor observado (FF, *Forward Filling*) o una interpolación lineal (LF, *Linear Filling*) entre la visit anterior y posterior. La tercera estrategia consiste en emplear las estimaciones del propio modelo (MF, *Model Filling*) para rellenar dichos datos.

Los datos utilizados son los proporcionados por el desafío TADPOLE, consistentes en 1677 sujetos de la base de datos ADNI.

2.7.1. Métodos

El modelo utilizado es un *minimalRNN*, cuya arquitectura se expone en la figura 2.12.

Las ecuaciones de actualización son las siguientes:

$$\begin{aligned}
 x_t &= [s_t, g_t] \\
 u_t &= \tanh(W_x x_t) \\
 f_t &= \sigma(U_h h_{t-1} + W_u u_t) \\
 h_t &= f_t \odot h_{t-1} + (1 - f_t) \odot u_t \\
 \hat{s}_{t+1} &= \text{softmax}(W_s h_t) \\
 \hat{g}_{t+1} &= W_g h_t + g_t
 \end{aligned}$$

Donde:

1. x_t es un vector con todas las variables observadas en el instante t .

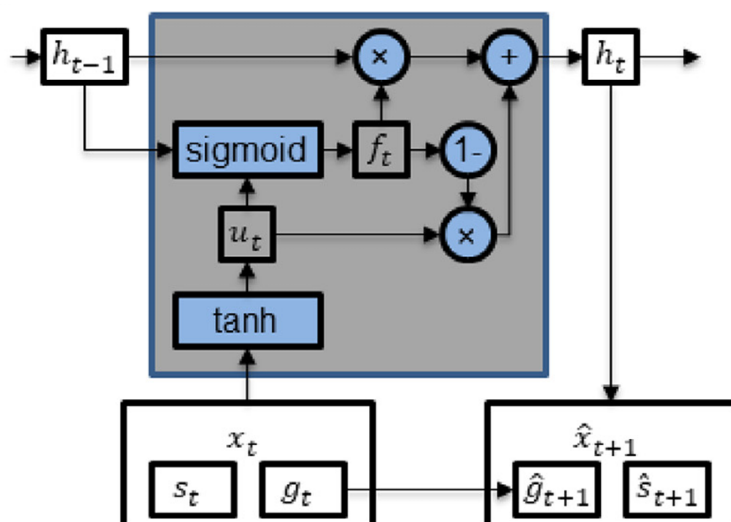


Figura 2.12: Arquitectura de *minimalRNN*. Fuente: Nguyen et al.[39]

2. s_t es el diagnóstico en el instante t . Se ha representado usando *one-hot encoding* con un vector de dimensión 3 (grupos CU, MCI, AD, respectivamente).
3. g_t es un vector (de dimensión 23) con los valores de los biomarcadores en el instante t .
4. u_t es el vector de entrada transformado mediante la tangente hiperbólica. De esta forma, los valores de entrada quedan comprendidos entre -1 y 1, y centrados en 0.
5. W_x son los pesos aplicados a x_t para obtener u_t .
6. f_t es la puerta de olvido (*forget gate*). Determina cuánto del estado anterior debe conservarse en el estado siguiente.
7. U_h son los pesos aplicados al estado h_{t-1} para calcular f_t .
8. W_u son los pesos aplicados a u_t para calcular f_t .
9. h_t es el estado oculto de la red en el instante t . Actúa como memoria interna, recordando información de estados anteriores.
10. W_s son los pesos aplicados a h_t para calcular \hat{s}_{t+1} .
11. W_g son los pesos aplicados a h_t para calcular \hat{g}_{t+1} .
12. \odot es el producto elemento a elemento.
13. σ es la función sigmoide.

La *minimalRNN* se actualiza usando el error entre los datos reales y los estimados según la siguiente función de pérdidas.

$$L = \sum_{t>1} (\text{CrossEntropy}(s_t, \hat{s}_t) + \text{MAE}(g_t, \hat{g}_t))$$

$$\text{CrossEntropy}(s_t, \hat{s}_t) = - \sum_{j=1}^3 s_t^j \log \hat{s}_t^j$$

$$\text{MAE}(g_t, \hat{g}_t) = \frac{1}{23} \sum_{j=1}^{23} |g_t^j - \hat{g}_t^j|$$

El optimizador utilizado es Adam (*Adaptive Moment Estimation*).

Las predicciones se realizan introduciendo como entradas las observaciones en cada visita. A partir del momento en el que no se disponga de más observaciones, se utiliza como entrada la estimación en el instante anterior.

El *set* de validación se usa para la selección de los hiperparámetros, usando el algoritmo HORD. El espacio de búsqueda se presenta en la tabla 2.6 para *minimalRNN*, LSS y LSTM, y la tabla 2.7 para SVM/SVR. La evaluación en validación y *test* se realiza usando la primera mitad de las observaciones de cada sujeto para predecir la segunda mitad. La división de los datos se repite 20 veces aleatoriamente para asegurar resultados estables.

Hiperparámetro	Rango
Tasa de <i>dropout</i> a la entrada	0.0 - 0.5
Tasa de <i>dropout</i> recurrente	0.0 - 0.5
Regularización de peso L2	10^{-7} - 10^{-5}
Tasa de aprendizaje	10^{-5} - 10^{-2}
Número de capas ocultas	1 - 3
Tamaño del estado oculto	128 - 512

Tabla 2.6: Espacio de búsqueda de hiperparámetros para *minimalRNN*, LSS y LSTM. Fuente: Nguyen et al.[39]

	SVM	SVR
Kernel	Lineal o RBF	
Epsilon	NA	10^{-3} - 10^{-0}
Penalización	10^{-3} - 10^3	
Gamma	10^{-3} - 10^3	

Tabla 2.7: Espacio de búsqueda de hiperparámetros para SVM/SVR. Fuente: Nguyen et al.[39]

2.7.2. Resultados

La figura 2.13 muestra el rendimiento de los modelos *minimalRNN-MF*, LSS-MF, LSTM-MF, predicción constante, y SVM (1 visita). Las medidas de calidad consideradas son el AUC, AUC, y MAE del ADAS13 y del volumen ventricular. RNN-MF (*minimalRNN-MF*) es el que mejor resultados presenta en todas las medidas.

Al separar los sujetos en grupos (sCU, pCU, sMCI, pMCI, rMCI, AD), RNN-MF sigue siendo superior a excepción de la predicción constante en el diagnóstico de sujetos estables. Si se analizan las predicciones anualmente hasta una distancia temporal de 6 años,

el rendimiento de todos los algoritmos desciende con la lejanía en el tiempo. La RNN-MF presenta los mejores resultados todos los años. Se observa también que el número de visitas no influye en los resultados de las predicciones, lo que sugiere que sería posible trabajar con datos transversales.

Aunque la RNN-MF presenta resultados significativamente mejores a otros modelos de referencia, un tratamiento efectivo para la enfermedad de Alzheimer debería comenzar al menos con una década de antelación. Dado que el rendimiento empeora notablemente con la distancia temporal, no es suficiente para su implantación clínica.

Por otro lado, el método MF es mejor que FF y LF a la hora de lidiar con datos ausentes.

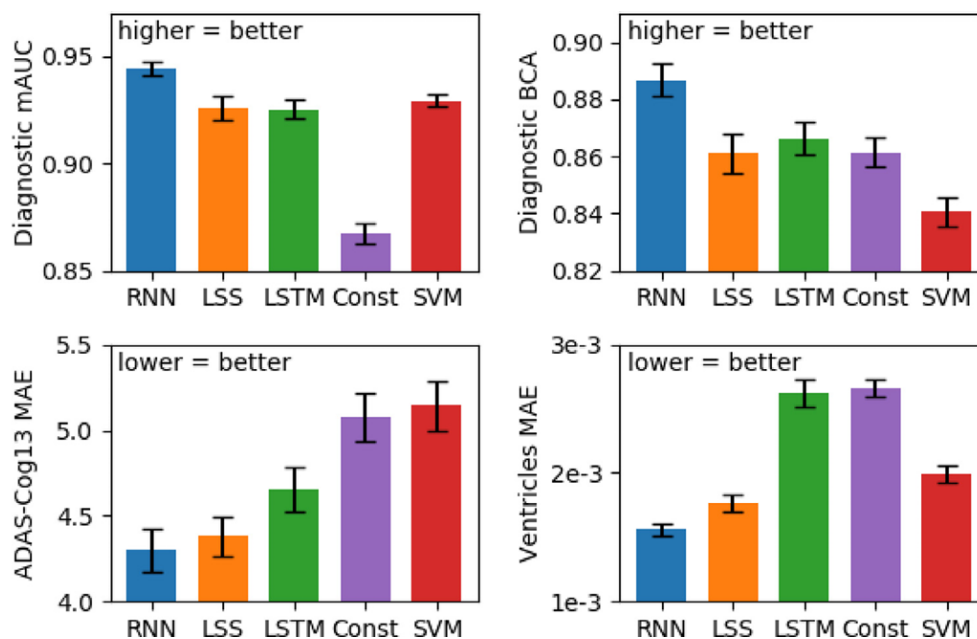


Figura 2.13: Resultados del AUC, AUC, y MAE del ADAS13 y del volumen ventricular de los modelos *minimalRNN-MF*, *LSS-MF*, *LSTM-MF*, predicción constante, y SVM (1 visita). Fuente: Nguyen et al.[39]

2.8. CARRNN: A Continuous Autoregressive Recurrent Neural Network for Deep Representation Learning From Sporadic Temporal Data

Diversas variantes de redes neuronales recurrentes (RNN) como *Long Short Term Memory* (LSTM) o *Gated Recurrent Unit* (LSTM) han sido propuestas para captar patrones temporales en datos longitudinales, evitando los problemas de desvanecimiento y explosión del gradiente. Sin embargo, las soluciones existentes no responden adecuadamente ante datos esporádicos (asíncronos e irregulares). Este artículo, combina RNNs con modelos autoregresivos de tiempo continuo (*Continuous-time AutoRegressive*, CAR), dando solución analítica al problema de la irregularidad en los datos. El CAR se implementa como una capa de la red neuronal, usando un modelo autorregresivo generalizado de tiempo discreto.

2.8.1. Métodos

El CARRNN aplica una RNN para aprender dependencias temporales a largo plazo y un CAR para compensar irregularidades en los datos.

Modelo CAR

Sea $\mathbf{x}_t \in \mathbb{R}^{N \times 1}$ un vector de características en el instante t predecible mediante un modelo autorregresivo multivariable de orden uno, donde el intervalo temporal entre observaciones puede variar:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{\Phi}\mathbf{x}(t) + \boldsymbol{\varsigma} + \mathbf{\Gamma}\frac{d\boldsymbol{\epsilon}(t)}{dt} \\ \mathbf{x}(t_k) &= e^{\mathbf{\Phi}\Delta t_k}\mathbf{x}(t_{k-1}) + \mathbf{\Phi}^{-1} [e^{\mathbf{\Phi}\Delta t_k} - \mathbf{I}_N] \boldsymbol{\varsigma} + \boldsymbol{\eta}(t_k) \\ \boldsymbol{\eta}(t_k) &= \int_{t_{k-1}}^{t_k} e^{(t_k-s)\mathbf{\Phi}}\mathbf{\Gamma}d\boldsymbol{\epsilon}(s) \end{aligned}$$

Aproximando el exponente matricial por series de potencias:

$$\begin{aligned} e^{\mathbf{\Phi}\Delta t_k} &= \sum_{p=0}^{\infty} \frac{(\mathbf{\Phi}\Delta t_k)^p}{p!} \approx \mathbf{I}_N + \mathbf{\Phi}\Delta t_k \\ \mathbf{x}(t_k) &\approx [\mathbf{I}_N + \mathbf{\Phi}\Delta t_k]\mathbf{x}(t_{k-1}) + \boldsymbol{\varsigma}\Delta t_k + \boldsymbol{\eta}(t_k) \end{aligned} \quad (2.1)$$

Modelo RNN

Se basa en las siguientes ecuaciones:

$$\begin{aligned} \mathbf{h}_t &= \sigma_h(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h\mathbf{h}_{t-1} + \mathbf{b}_h) \\ \mathbf{y}_t &= \sigma_y(\mathbf{W}_y\mathbf{h}_t + \mathbf{b}_y) \end{aligned}$$

Donde $\mathbf{h}_t \in \mathbb{R}^{M \times 1}$ e $\mathbf{y}_t \in \mathbb{R}^{Q \times 1}$ son las capas oculta y de salida, respectivamente, separadas por un tiempo uniforme t ; $\mathbf{W}_h \in \mathbb{R}^{M \times N}$, $\mathbf{U}_h \in \mathbb{R}^{M \times M}$, $\mathbf{b}_h \in \mathbb{R}^{M \times 1}$, $\mathbf{W}_y \in \mathbb{R}^{Q \times M}$ y $\mathbf{b}_y \in \mathbb{R}^{Q \times 1}$ son parámetros ajustables del modelo; σ_h y σ_y son funciones de activación.

Modelo CARRNN

Integrando el modelo CAR con modelos RNN, LSTM o LSTM se obtienen CAR-RNN, CAR-LSTM y CAR-GRU (figura 2.14):

- CAR-RNN:

$$\begin{aligned} \tilde{\mathbf{h}}_k &= \sigma_h(\mathbf{W}_h\mathbf{x}_k + \mathbf{U}_h\mathbf{h}_{k-1} + \mathbf{b}_h) \\ \mathbf{h}_k &= [\mathbf{I}_M + (\Delta t_k - \tau)\mathbf{\Phi}_h] \tilde{\mathbf{h}}_k + (\Delta t_k - \tau)\boldsymbol{\varsigma}_h \end{aligned} \quad (2.2)$$

- CAR-LSTM:

$$\tilde{\mathbf{f}}_k = \sigma_g(\mathbf{W}_f\mathbf{x}_k + \mathbf{U}_f\mathbf{h}_{k-1} + \mathbf{V}_f\mathbf{c}_{k-1} + \mathbf{b}_f)$$

$$\begin{aligned}
\tilde{\mathbf{i}}_k &= \sigma_g(\mathbf{W}_i \mathbf{x}_k + \mathbf{U}_i \mathbf{h}_{k-1} + \mathbf{V}_i \mathbf{c}_{k-1} + \mathbf{b}_i) \\
\tilde{\mathbf{z}}_k &= \sigma_c(\mathbf{W}_z \mathbf{x}_k + \mathbf{U}_z \mathbf{h}_{k-1} + \mathbf{b}_z) \\
\tilde{\mathbf{c}}_k &= \sigma_h(\tilde{\mathbf{f}}_k \odot \mathbf{c}_{k-1} + \tilde{\mathbf{i}}_k \odot \tilde{\mathbf{z}}_k) \\
\tilde{\mathbf{o}}_k &= \sigma_g(\mathbf{W}_o \mathbf{x}_k + \mathbf{U}_o \mathbf{h}_{k-1} + \mathbf{V}_o \mathbf{c}_k + \mathbf{b}_o) \\
\tilde{\mathbf{h}}_k &= \tilde{\mathbf{o}}_k \odot \tilde{\mathbf{c}}_k
\end{aligned}$$

Donde \mathbf{h}_k y \mathbf{c}_k pueden obtenerse de forma similar a la mencionada en (2.2); $\{\tilde{\mathbf{f}}_k, \tilde{\mathbf{i}}_k, \tilde{\mathbf{z}}_k, \tilde{\mathbf{c}}_k, \tilde{\mathbf{o}}_k, \tilde{\mathbf{h}}_k\} \in \mathbb{R}^{M \times 1}$ son los vectores de la puerta de olvido, puerta de entrada, puerta de modulación, estado de la celda, puerta de salida y estado oculto; $\{\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_z, \mathbf{W}_o\} \in \mathbb{R}^{M \times N}$, $\{\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_z, \mathbf{U}_o\} \in \mathbb{R}^{M \times M}$, $\{\mathbf{V}_f, \mathbf{V}_i, \mathbf{V}_o\} \in \mathbb{R}^{M \times M}$, $\{\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_z, \mathbf{b}_o\} \in \mathbb{R}^{M \times 1}$ son parámetros ajustables de la red; $\sigma_g, \sigma_c, \sigma_h$ son las funciones de activación de las puertas de entrada, modulación y salida.

■ CAR-GRU:

$$\begin{aligned}
\tilde{\mathbf{z}}_k &= \sigma_g(\mathbf{W}_z \mathbf{x}_k + \mathbf{U}_z \mathbf{h}_{k-1} + \mathbf{b}_z) \\
\tilde{\mathbf{r}}_k &= \sigma_g(\mathbf{W}_r \mathbf{x}_k + \mathbf{U}_r \mathbf{h}_{k-1} + \mathbf{b}_r) \\
\tilde{\mathbf{c}}_k &= \sigma_h(\mathbf{W}_c \mathbf{x}_k + \mathbf{U}_c(\tilde{\mathbf{r}}_k \odot \mathbf{h}_{k-1}) + \mathbf{b}_c) \\
\tilde{\mathbf{h}}_k &= (1 - \tilde{\mathbf{z}}_k) \odot \tilde{\mathbf{c}}_k + \tilde{\mathbf{z}}_k \odot \mathbf{h}_{k-1}
\end{aligned}$$

Donde \mathbf{h}_k puede obtenerse de forma similar a la mencionada en (2.2); $\{\tilde{\mathbf{z}}_k, \tilde{\mathbf{r}}_k, \tilde{\mathbf{c}}_k, \tilde{\mathbf{h}}_k\} \in \mathbb{R}^{M \times 1}$ son los vectores de la puerta de actualización, puerta de reseteo, estado candidato y estado oculto; $\{\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_c\} \in \mathbb{R}^{M \times N}$, $\{\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_c\} \in \mathbb{R}^{M \times M}$, $\{\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_c\} \in \mathbb{R}^{M \times 1}$ son parámetros ajustables de la red.

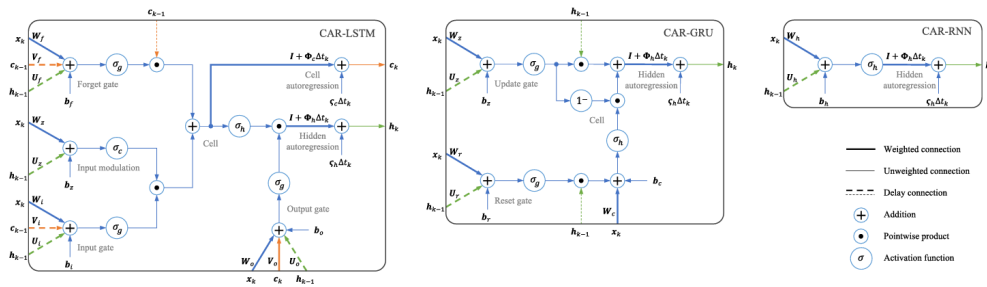


Figura 2.14: Arquitecturas de los modelos CAR-LSTM, CAR-GRU y CAR-RNN. Fuente: Ghazi et al.[32]

Time binning y valores faltantes

El *time binning* agrupa características en intervalos (*bins*) de anchura igual al paso temporal τ de la RNN. De esta forma, se consigue el alineamiento temporal de los datos (figura 2.15).

Durante el entrenamiento, el modelo CARRNN se encarga de imputar valores faltantes de las características, mediante una versión univariante de (2.1):

$$x_{n,k} = [1 + (t_k - t_j)\varphi(n)] x_{n,j} + (t_k - t_j)\zeta(n)$$

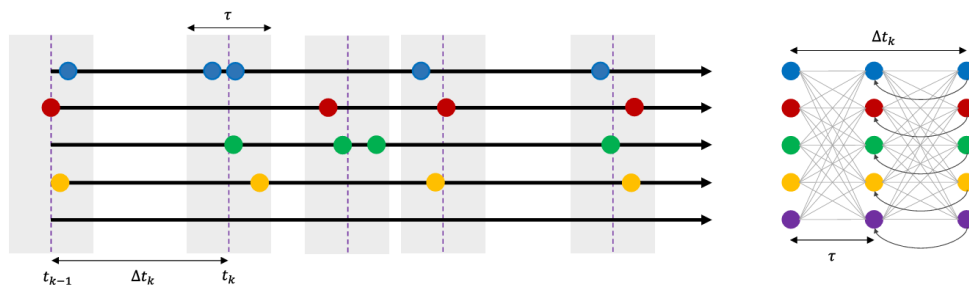


Figura 2.15: Series temporales con eventos asíncronos y datos separados por intervalos de tiempo irregulares, y estructura CARRNN propuesta para modelar los datos agrupados. La figura de la izquierda muestra 5 secuencias de características alineadas por intervalos de anchura τ y una separación de $\Delta t_k = t_k - t_{k-1}$. La figura de la derecha muestra el modelo CARRNN que aprende dependencias temporales multivariantes a partir de datos alineados. Fuente: Ghazi et al.[32]

Donde $\varphi \in \mathbb{R}^{N \times 1}$ y $\zeta \in \mathbb{R}^{N \times 1}$ son los parámetros del CAR univariante; $x_{n,k}$ es el valor de la n -ésima característica en el instante t_k basado en la observación vecina en el instante t_j .

En el caso de tener valores faltantes al inicio de la secuencia o características totalmente vacías a la entrada y salida, se utilizan un vector de entrada y una función de pérdida ponderados según se explica en la figura 2.16.

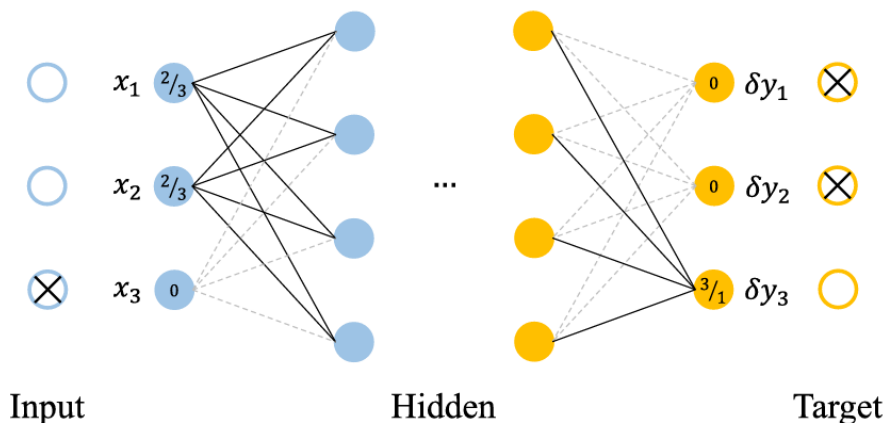


Figura 2.16: Entrenamiento con valores faltantes utilizando un proceso ponderado de propagación hacia delante (*feedforward*) y retropropagación (*backpropagation*). En el ejemplo, $Q=N=3$ y hay un valor de entrada y dos de salida faltantes marcados con cruces. Los nodos y pesos asociados con el valor de entrada faltante y los nodos de salida se fijan a 0. Los valores de entrada conocidos se escalan por un factor de $2/3$ (*inputs* disponibles entre *inputs* totales, N) y los de salida por $3/1$ (*outputs* totales, Q , entre *outputs* disponibles). Fuente: Ghazi et al.[32]

2.8.2. Resultados

Los modelos CARRNN se aplicaron a problemas de regresión y clasificación, usando como funciones de activación la identidad, tangente hiperbólica y sigmoide para σ_h , σ_c y σ_g , respectivamente. Se utilizaron 4 *sets* de datos reales (ADNI, PhysioNet/CinC, N-MNIST, datos de actividad de personas de la UCI) y 1 con datos sintéticos. Las características se estandarizaron y los intervalos de tiempo se normalizaron con el rango intercuartílico

(IQR).

Los modelos utilizados en los experimentos fueron: *GRU-Mean*, *GRU-Forward*, *GRU-Concat*, *GRU-D*, *PLSTM*, *ODE-RNN*, *CAR*, *CAR-RNN*, *CAR-LSTM* y *CAR-GRU*. Se usaron la norma L2 y la entropía cruzada como funciones de pérdida en tareas de regresión y clasificación, respectivamente. Se eligió el optimizador Adam. Las redes fueron entrenadas 1000 épocas con una validación cruzada de 10 particiones y usando *early-stopping*.

Se obtuvieron varias conclusiones:

- CAR-GRU presenta el mejor desempeño tanto en tareas de regresión como de clasificación.
- El tamaño óptimo de los *bins* se obtiene próximo a la media de los intervalos entre los datos en tareas de regresión y cercano al cuartil superior en tareas de clasificación.
- La arquitectura propuesta es generalizable a cualquier modelo en tiempo discreto o continuo. Además, el CAR es fácilmente integrable en otros marcos de *deep learning*, incluso CNN (*Convolutional Neural Networks*).

2.9. Deep recurrent model for individualized prediction of Alzheimer’s disease progression

Los DPM basados en modelos de aprendizaje profundo presentan la ventaja de aprender directamente de los datos en vez de pasar por un proceso previo de ingeniería de características. Las RNN, en concreto las LSTM, son ideales cuando los datos son longitudinales debido a que permiten resolver metodológicamente problemas de alineación temporal. Sin embargo, su uso en DPM sigue siendo complicado debido a la incompletitud de las visitas. Numerosos estudios previos han resuelto esta situación en un paso previo de preprocesamiento o como una subtarea durante el entrenamiento del modelo.

Jung et al. proponen un modelo que dinámicamente imputa los valores faltantes; utiliza una LSTM para codificar las observaciones en una representación latente; predice las medidas volumétricas de MRI, las trayectorias de los marcadores cognitivos y el estado clínico a lo largo del tiempo. De forma metodológica, aprovechan las relaciones temporales y multivariantes entre las variables para estimar los valores faltantes. Además, los parámetros de los módulos (imputación, codificación, predicción) que componen el modelo se ajustan de forma conjunta.

2.9.1. Métodos

El modelo se compone de 3 módulos (imputación, codificación y predicción), representados en la figura 2.17. El módulo de imputación estima los valores faltantes usando la representación latente de la visita anterior y las relaciones multivariantes entre características. El módulo de codificación codifica la observación imputada y la información de diferencias de tiempo entre observaciones en una representación latente que captura las dependencias temporales. El módulo de predicción estima las características MRI y cognitivas de la siguiente visita y predice el estado clínico de la visita actual basándose en la representación latente.

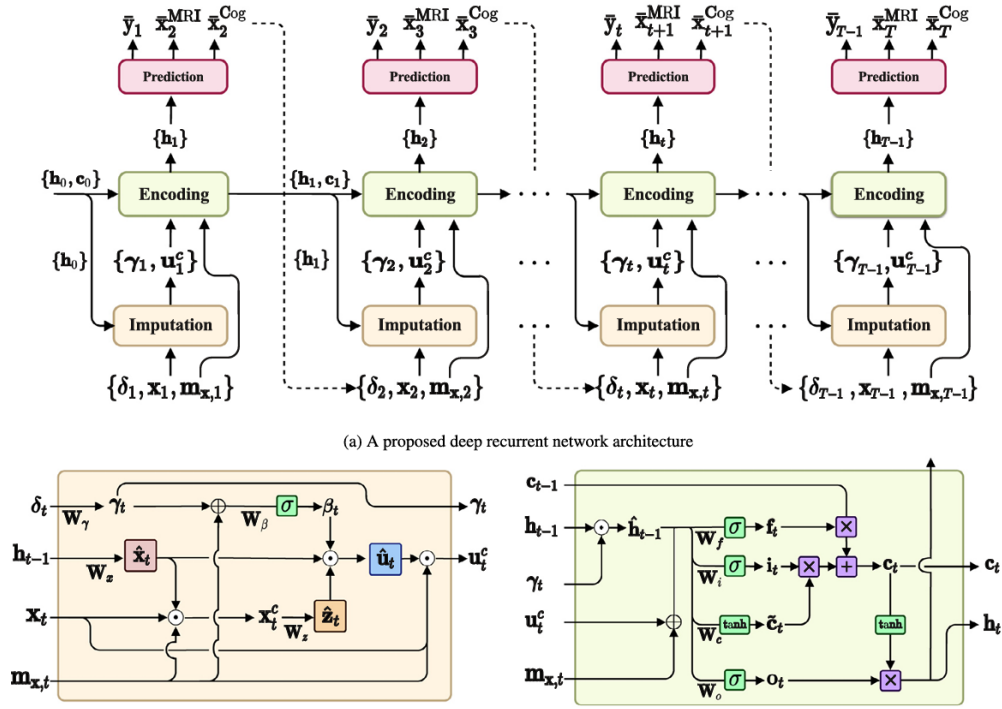


Figura 2.17: (a) Arquitectura del modelo propuesto combinando los módulos de imputación, codificación y predicción. Los módulos estiman los valores faltantes en la entrada actual, predicen los marcadores MRI, cognitivos de la visita siguiente y el estado clínico de la visita actual. (b) Operaciones para considerar las relaciones temporales y entre características en la imputación de valores faltantes. (c) Variante de la unidad LSTM estándar incluyendo γ y $m_{x,t}$. Por simplicidad se han omitido los sesgos. Si no hay observaciones, los valores predichos por el modelo (representados con flechas discontinuas) se usan como entrada del siguiente instante de tiempo. Las flechas continuas indican el funcionamiento de la arquitectura. Fuente: Jung et al.[36]

Imputación de valores faltantes

Las ecuaciones que describen el módulo de imputación se presentan a continuación:

$$\begin{aligned}
 \hat{x}_t &= f_x(\mathbf{h}_{t-1}) = \mathbf{W}_x \mathbf{h}_{t-1} + \mathbf{b}_x \\
 \mathbf{x}_t^c &= \mathbf{m}_{x,t} \odot \mathbf{x}_t + (\mathbf{1} - \mathbf{m}_{x,t}) \odot \hat{x}_t \\
 \hat{z}_t &= \mathbf{W}_z \mathbf{x}_t^c + \mathbf{b}_z \\
 \gamma_x &= \exp\{-\max(\mathbf{0}, \mathbf{W}_{\gamma_x}^\top \delta_t + \mathbf{b}_{\gamma_x})\} \\
 \beta_t &= \sigma(\mathbf{W}_\beta \odot [\gamma_x \oplus \mathbf{m}_{x,t}] + \mathbf{b}_\beta) \\
 \hat{u}_t &= \beta_t \odot \hat{z}_t + (\mathbf{1} - \beta_t) \odot \hat{x}_t \\
 \mathbf{u}_t^c &= \mathbf{m}_{x,t} \odot \mathbf{x}_t + (\mathbf{1} - \mathbf{m}_{x,t}) \odot \hat{u}_t \\
 \Delta(d, t) &= \begin{cases} s(t) - s(t-1) + \Delta(d, t-1), & t > 1 \wedge \mathbf{M}_x(d, t-1) = 0 \\ s(t) - s(t-1), & t > 1 \wedge \mathbf{M}_x(d, t-1) = 1 \\ 1, & t = 1 \end{cases}
 \end{aligned}$$

Donde $f_x(\mathbf{h}_{t-1})$ representa la red de propagación hacia adelante; $\{\mathbf{W}_x, \mathbf{W}_z, \mathbf{W}_{\gamma_t}, \mathbf{W}_{\gamma_x}, \mathbf{W}_\beta, \mathbf{b}_x, \mathbf{b}_z, \mathbf{b}_{\gamma_t}, \mathbf{b}_{\gamma_x}, \mathbf{b}_\beta\}$ son parámetros ajustables de la red; \mathbf{h}_t es el estado oculto en la visita t ; \mathbf{M}_x y \mathbf{m}_x son las máscaras de las características de entrada y de los estados clínicos; Δ , $\Delta(d, t)$ y δ_t son la matriz de diferencias temporales entre observaciones, la d -ésima variable de dicha matriz en la visita t y la matriz en la visita t , respectivamente.

Primero, se estiman los marcadores, $\hat{\mathbf{x}}_t$, a partir del estado oculto, que codifica las dependencias temporales. Segundo, se imputan los valores faltantes, dejando inalterados los disponibles, y dando lugar a \mathbf{x}_t^c . Tercero, se tienen en cuenta las relaciones multivariantes en $\hat{\mathbf{z}}_t$. Cuarto, se construye el vector de coeficientes de ponderación, β_t , que tiene en cuenta tanto las diferencias de tiempo entre observaciones a través del factor de disminución temporal, γ_x , como los valores faltantes en la visita actual. El factor de disminución temporal queda comprendido entre 0 y 1 y representa la lejanía de la anterior observación para cada variable. Quinto, se integra la información del vector de coeficientes de ponderación en $\hat{\mathbf{u}}_t$. De esta forma, se pondera la influencia entre las relaciones temporales y multivariantes. Sexto, solamente los valores faltantes son imputados, resultando en \mathbf{u}_t^c .

Codificación temporal recurrente y predicción multitarea

Las ecuaciones que describen los módulos de codificación y de predicción se presentan a continuación:

$$\begin{aligned}
\gamma_t &= \exp\{-\text{máx}(\mathbf{0}, \mathbf{W}_{\gamma_t}^\top \delta_t + \mathbf{b}_{\gamma_t})\} \\
\hat{\mathbf{h}}_{t-1} &= \mathbf{h}_{t-1} \odot \gamma_t \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_{fo} \odot \left[\hat{\mathbf{h}}_{t-1}, \mathbf{u}_t^c \oplus \mathbf{m}_{x,t}\right] + \mathbf{b}_{fo}\right) \\
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i \odot \left[\hat{\mathbf{h}}_{t-1}, \mathbf{u}_t^c \oplus \mathbf{m}_{x,t}\right] + \mathbf{b}_i\right) \\
\tilde{\mathbf{c}}_t &= \tanh\left(\mathbf{W}_c \odot \left[\hat{\mathbf{h}}_{t-1}, \mathbf{u}_t^c \oplus \mathbf{m}_{x,t}\right] + \mathbf{b}_c\right) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o \odot \left[\hat{\mathbf{h}}_{t-1}, \mathbf{u}_t^c \oplus \mathbf{m}_{x,t}\right] + \mathbf{b}_o\right) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \\
\tilde{\mathbf{x}}_{t+1}^{MRI} &= \mathbf{W}_{MRI} \mathbf{h}_t + \mathbf{b}_{MRI} \\
\tilde{\mathbf{x}}_{t+1}^{Cog} &= \mathbf{W}_{Cog} \mathbf{h}_t + \mathbf{b}_{Cog} \\
\tilde{\mathbf{y}}_t &= \text{softmax}(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y)
\end{aligned}$$

Donde $\{\mathbf{W}_{fo}, \mathbf{W}_i, \mathbf{W}_c, \mathbf{W}_o, \mathbf{W}_{MRI}, \mathbf{W}_{Cog}, \mathbf{W}_y, \mathbf{b}_{fo}, \mathbf{b}_i, \mathbf{b}_c, \mathbf{b}_o, \mathbf{b}_{MRI}, \mathbf{b}_{Cog}, \mathbf{b}_y\}$ son parámetros ajustables de la red. Los vectores de pesos se dividen en 2 vectores internos, $\mathbf{W}_{(gate)i}$ y $\mathbf{W}_{(gate)h}$, conectados a las entradas y a la información temporal, respectivamente. Las operaciones con dichos vectores se realizarían de la siguiente forma:

$$\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{W}_{fo} \odot \left[\hat{\mathbf{h}}_{t-1}, \mathbf{u}_t^c \oplus \mathbf{m}_{x,t}\right] + \mathbf{b}_{fo}\right) \\
&= \sigma\left(\mathbf{W}_{foh} \odot \hat{\mathbf{h}}_{t-1} + \mathbf{b}_{foh} + \mathbf{W}_{foi} \odot (\mathbf{u}_t^c \oplus \mathbf{m}_{x,t}) + \mathbf{b}_{foi}\right)
\end{aligned}$$

Primero, se calcula el factor de disminución temporal, γ_t . La diferencia entre γ_t y γ_x es que la dimensión es igual al número de unidades ocultas y de variables, respectivamente. Segundo, se determina $\hat{\mathbf{h}}_{t-1}$, que tiene en cuenta cuándo se observó el último dato real. Tercero, se calculan las puertas de olvido, \mathbf{f}_t , y de entrada, \mathbf{i}_t . La primera decide qué información eliminar, mientras que la segunda propone información a añadir a través del estado candidato, $\tilde{\mathbf{c}}_t$. Cuarto, se actualiza el estado de la celda, \mathbf{c}_t . Quinto, se calcula la puerta de salida, \mathbf{o}_t , que protege a las siguientes celdas de información irrelevante y filtra

la información a considerar. Sexto, se actualiza el estado oculto, \mathbf{h}_t . Séptimo, se predicen las salidas (MRI, marcadores cognitivos, estado clínico): $\tilde{\mathbf{x}}_{t+1}^{MRI}$, $\tilde{\mathbf{x}}_{t+1}^{Cog}$, $\tilde{\mathbf{y}}_t$.

Entrenamiento colaborativo

Se define una función de pérdidas global para el entrenamiento conjunto de todos los módulos:

$$\begin{aligned}\mathcal{L}_{imputation} &= \sum_{t=1}^T (|\tilde{\mathbf{x}}_t - \hat{\mathbf{x}}_t| + |\tilde{\mathbf{x}}_t - \hat{\mathbf{z}}_t| + |\tilde{\mathbf{x}}_t - \hat{\mathbf{u}}_t|) \odot (1 - \tilde{\mathbf{m}}_{x,t}) \\ \mathcal{L}_{MRI} &= \sum_{t=1}^{T-1} (\mathbf{x}_{t+1}^{MRI} \odot \mathbf{m}_{x,t+1} - \tilde{\mathbf{x}}_{t+1}^{MRI} \odot \mathbf{m}_{x,t+1})^2 \\ \mathcal{L}_{Cog} &= \sum_{t=1}^{T-1} (\mathbf{x}_{t+1}^{Cog} \odot \mathbf{m}_{x,t+1} - \tilde{\mathbf{x}}_{t+1}^{Cog} \odot \mathbf{m}_{x,t+1})^2 \\ \mathcal{L}_y &= - \sum_{t=1}^{T-1} \mathbf{m}_{y,t} \left[\sum_{k=1}^K \mathbf{y}_t(k) (1 - \tilde{\mathbf{y}}_t(k))^\epsilon \log(\tilde{\mathbf{y}}_t(k)) \right] \\ \mathcal{L}_{total} &= \alpha \mathcal{L}_{imputation} + \zeta (\mathcal{L}_{MRI} + \mathcal{L}_{Cog}) + \xi \mathcal{L}_y\end{aligned}$$

Donde $\tilde{\mathbf{x}}_t = \mathbf{x}_t \odot \mathbf{m}_{x,t}$ y $\tilde{\mathbf{m}}_{x,t}$ es una matriz de eliminación de valores aleatorios; $\{\epsilon, \alpha, \zeta, \xi\}$ son hiperparámetros.

$\mathcal{L}_{imputation}$ mide la similaridad mediante el MAE entre los datos imputados y los reales teniendo en cuenta relaciones temporales, multivariantes y combinadas. Durante el entrenamiento, los valores a imputar se eliminan de forma aleatoria. \mathcal{L}_{MRI} y \mathcal{L}_{Cog} usan el MSE entre los valores reales de los marcadores MRI y cognitivos, y los estimados. \mathcal{L}_y es la entropía cruzada focal para la predicción del estado clínico.

2.9.2. Resultados

Los datos utilizados son los proporcionados por el desafío TADPOLE, excluyendo aquellos con menos de 3 visitas. Se consideran características volumétricas (ventrículos, hipocampo, giro fusiforme, giro temporal medio, corteza entorrinal y cerebro completo), cognitivas (MMSE, ADAS11, ADAS13) y el estado clínico (CN, MCI, AD). Los marcadores MRI se normalizaron con el volumen intracraneal del sujeto (ICV) y entre -1 y 1, mientras que los marcadores cognitivos se normalizaron entre 0 y 1.

Se realizaron predicciones hasta los 11 instantes siguientes al inicio del estudio. El espacio de búsqueda de hiperparámetros consistió en $\{5 \times 10^{-5}, 5 \times 10^{-4}, 5 \times 10^{-3}, 5 \times 10^{-2}\}$ para la tasa de aprendizaje, $\{1, 2, 3\}$ para las capas ocultas, $\{12, 32, 48, 64, 80, 96\}$ para las unidades ocultas, $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ para la regularización L2. Se utilizó un proceso de validación cruzada con 5 particiones. La tarea de imputación se evaluó con el MAE y el MRE. En la predicción de marcadores MRI y cognitivos se utilizó el MAE y el RMSE, respectivamente. En la predicción del estado clínico se utilizó mAUC, sensibilidad y precisión. Los resultados se compararon con los de los siguientes modelos: *Mean imputation with LSTM* (LSTM-M), *Forward imputation with LSTM* (LSTM-FF), *Zero imputation with Peephole LSTM* (PLSTM-Z), *Multi-directional RNN* (MRNN), *Model filling with MinimalRNN* (MinimalRNN). La significancia estadística con dichos métodos

se evaluó con la prueba bilateral pareada Wilcoxon de rangos con signo y el coeficiente de correlación de Pearson para tareas de regresión, y la prueba de McNemar para tareas de clasificación.

El método propuesto logró los resultados más bajos de MAE y MRE con significancia estadística en la imputación. Igualmente, obtuvo las mejores puntuaciones en la estimación de los marcadores MRI y cognitivos. Lo mismo ocurrió con el mAUC y la precisión. Al predecir el estado clínico considerando el número de visitas, el mAUC se mostró elevado y estable en todos los casos. Además, los métodos de imputación integrativos mostraron mejores resultados que los basados en preprocesamiento. Por otro lado, el modelo propuesto resultó menos propenso a predecir sujetos reversores que otros del estado del arte, lo que se alinea mejor con la irreversibilidad de la enfermedad.

Adicionalmente, se llevaron a cabo estudios de ablación:

- Efecto del aprendizaje multitarea: al entrenar el modelo para predecir separadamente los marcadores MRI y cognitivos, y utilizar un clasificador independiente basado en LDA, se obtuvieron peores resultados que en el entrenamiento colaborativo.
- Valores óptimos de α , ζ , ξ y ϵ : se obtuvo 0.1, 0.5, 0.5 y 5, respectivamente.
- Efecto de la información sobre los valores faltantes: los peores resultados se obtuvieron al ignorar las máscaras y las diferencias de tiempo entre observaciones. Su efecto fue incluso mayor que el de las relaciones multivariantes entre características.

2.10. Multi-modal sequence learning for Alzheimer’s disease progression prediction with incomplete variable-length longitudinal data

A diferencia de los métodos estadísticos, los modelos de aprendizaje automático no requieren tantas suposiciones, y se dividen en 3 categorías: métodos para series temporales, métodos multitarea y métodos de aprendizaje profundo. Los primeros exigen una parametrización de las trayectorias longitudinales, la cual podría desviarse de la trayectoria específica de los sujetos. Los segundos quedan restringidos a individuos con un número fijo de visitas, además de incluir términos de regularización sofisticados. Los últimos capturan dependencias temporales de largo plazo, salvando los obstáculos presentes en las anteriores categorías.

Adicionalmente, la mayoría de estudios realizados hasta el momento construyen los DPM según una modalidad de datos. Sin embargo, la enfermedad de Alzheimer comprende múltiples modalidades complementarias (p.ej, PET, MRI o datos demográficos). La combinación de varias modalidades, teniendo en cuenta la información complementaria que ofrece cada una, podría reflejar con más precisión la progresión de la enfermedad.

Finalmente, los datos faltantes (p.ej, algunas pruebas no se realizan por su alto coste, o los pacientes no aparecen en la fecha acordada, o incluso abandonan el estudio) es un problema recurrente que afecta de forma negativa en el entrenamiento de los modelos. Una solución posible es abordar el problema en una etapa de preprocesamiento en la que se imputan estos datos, y posteriormente llevar a cabo el entrenamiento. Otra opción es integrar la imputación como parte del entrenamiento.

Teniendo en cuenta todo lo anterior, Xu et al. proponen un modelo de aprendizaje profundo capaz de manejar secuencias de longitud variable, imputando datos faltantes de forma arbitraria, y de explotar la correlación entre diversas modalidades. El modelo combina un módulo de fusión de modalidades con un módulo de aprendizaje de secuencias basado en RNN. El primero se encarga de aprender la representación latente subyacente compartida por las modalidades, de la cual se originan. El segundo procesa secuencias de longitud variable y modela la progresión de la enfermedad.

2.10.1. Métodos

El modelo se compone de un módulo de fusión de modalidades y un módulo de aprendizaje de secuencias basado en RNN, esquematizado en la figura 2.18.

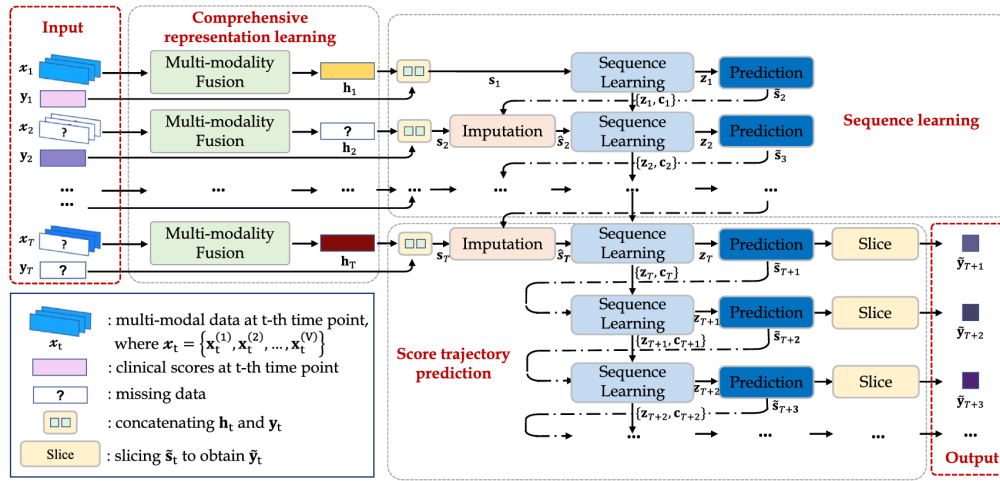


Figura 2.18: Representación del modelo propuesto. El módulo de fusión de modalidades utiliza la información longitudinal multimodal para aprender la representación latente. El módulo de aprendizaje de secuencias estima en cada paso los valores faltantes. En tareas de predicción, se usan los valores estimados en la visita anterior como entrada actual. Fuente: Xu et al.[50]

Módulo de fusión de modalidades

El módulo de fusión de modalidades se encarga de aprender la representación latente subyacente compartida por las modalidades (en este caso, PET, MRI, datos demográficos). Se considera que cada modalidad representa al individuo desde un punto de vista diferente. Por lo tanto, la representación latente debe ser una representación completa que incluye información de todas las modalidades. Partiendo de dicha representación, los datos de las modalidades se reconstruyen utilizando funciones de degradación. La implementación del módulo se realiza mediante redes de degradación, según se muestra en la figura 2.19. Las visitas con alguna modalidad faltante, serán reconstruidas en base a la representación latente. La información relativa a visitas faltantes puede obtenerse basándose en los datos demográficos (lugar, edad, género, educación, $APOE \epsilon 4$).

La función de pérdida del módulo de fusión de modalidades viene dada por:

$$\mathcal{L}_{rec} = \sum_{v=1}^V \mathcal{L}_v = \sum_{v=1}^V \sum_{i=1}^N \sum_{t=1}^T \left\| o_{i,t}^{(v)} \left(f_v(\mathbf{h}_{i,t}; \Theta_v) - \mathbf{x}_{i,t}^{(v)} \right) \right\|_2^2$$

Donde $f_v(\cdot; \Theta_v)$ es la red de degradación de la v -ésima modalidad de parámetros Θ_v ; $\mathbf{x}_{i,t}^{(v)}$ son los datos de la v -ésima modalidad del i -ésimo individuo en el instante t ; $o_{i,t}^{(v)}$ indica si el i -ésimo individuo tiene datos para la v -ésima modalidad en el instante t ($o_{i,t}^{(v)} = 1$) o no ($o_{i,t}^{(v)} = 0$).

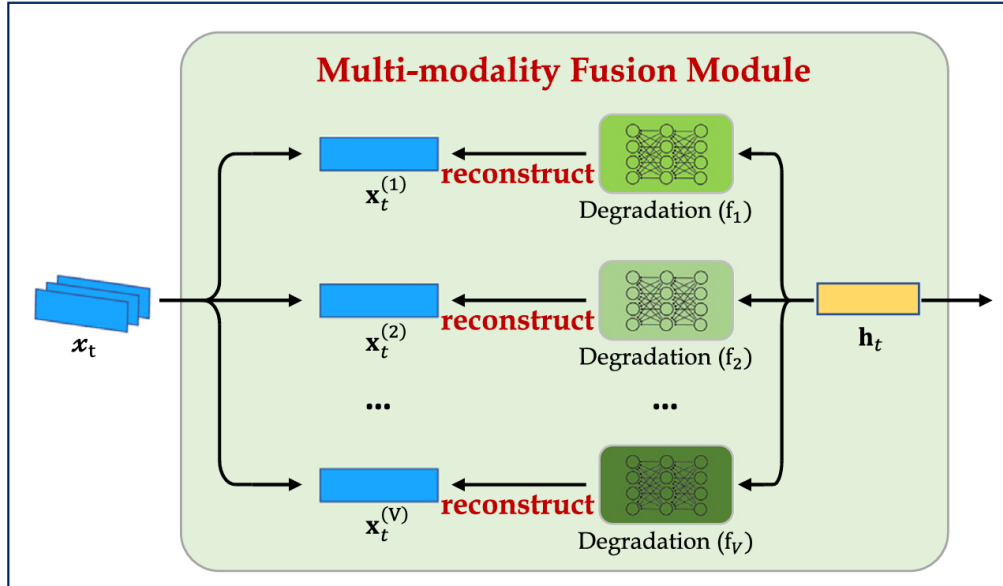


Figura 2.19: Representación del módulo de fusión de modalidades, donde $\mathbf{x}_t^{(v)}$ son las características de la v -ésima modalidad y \mathbf{h}_t es la representación latente en el instante t para el individuo i . “Degradation (f_v)” es la red de degradación de la v -ésima modalidad. Fuente: Xu et al.[50]

Módulo de aprendizaje de secuencias basado en RNN

El módulo de aprendizaje de secuencias utiliza una LSTM (*Long Short Term Memory*), cuya estructura se muestra en la figura 2.20. A continuación, se presentan las ecuaciones de la LSTM:

$$\begin{aligned}
 \mathbf{f}_t &= \sigma([\mathbf{z}_{t-1}, \hat{\mathbf{s}}_t] \mathbf{W}_f + \mathbf{b}_f) \\
 \mathbf{i}_t &= \sigma([\mathbf{z}_{t-1}, \hat{\mathbf{s}}_t] \mathbf{W}_i + \mathbf{b}_i) \\
 \mathbf{o}_t &= \sigma([\mathbf{z}_{t-1}, \hat{\mathbf{s}}_t] \mathbf{W}_o + \mathbf{b}_o) \\
 \tilde{\mathbf{c}}_t &= \tanh([\mathbf{z}_{t-1}, \hat{\mathbf{s}}_t] \mathbf{W}_c + \mathbf{b}_c) \\
 \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
 \mathbf{z}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
 \end{aligned}$$

Donde $\hat{\mathbf{s}}_t$, \mathbf{c}_t , $\tilde{\mathbf{c}}_t$ y \mathbf{z}_t son la entrada actual, el estado de la celda, el estado candidato y el estado oculto en la visita t , respectivamente; $\{\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c, \mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_c\}$ son parámetros de la red. La entrada de la red corresponde a la concatenación de la representación latente y la salida: $\mathbf{s}_t = [\mathbf{h}_t, \mathbf{y}_t]$.

Para abordar el problema de los valores faltantes se utiliza una estrategia de “*Model Filling*”. Es decir, en cada visita el modelo estima el valor de la entrada basándose en el estado oculto de la visita anterior. Los valores faltantes se imputan con las estimaciones del

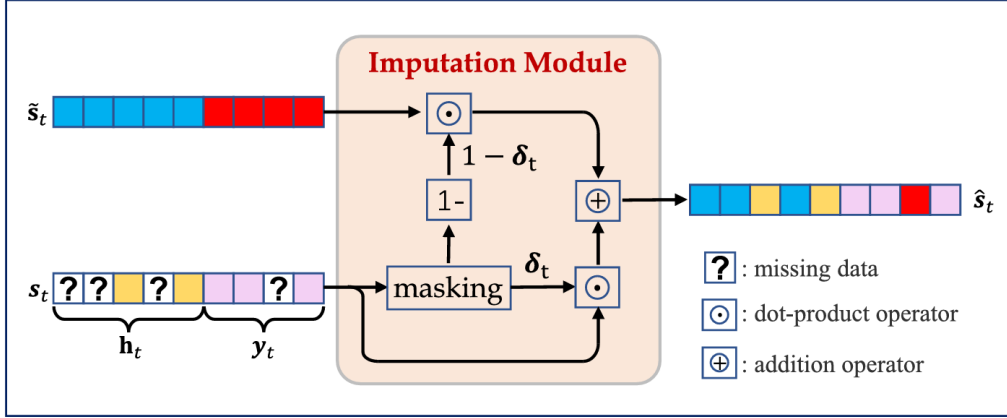


Figura 2.21: Representación del módulo de imputación, donde $\mathbf{s}_t = [\mathbf{h}_t, \mathbf{y}_t]$. La máscara δ_t indica los valores faltantes en \mathbf{s}_t , que serán imputados con el correspondiente valor de $\tilde{\mathbf{s}}_t$. Fuente: Xu et al.[50]

Donde $\mathcal{F}(\cdot)$ es el operador *slice* sobre $\tilde{\mathbf{s}}_{i,t}$; $\mathbf{q}_{i,t}$ es un indicador tal que $q_{(i,j),t} = 1$ si la j -ésima característica del i -ésimo individuo en la visita t existe y $q_{(i,j),t} = 0$ si es faltante.

Entrenamiento colaborativo

En vez de entrenar los módulos anteriores por separado, se lleva a cabo un entrenamiento colaborativo. En otras palabras, el resultado de la predicción de la LSTM se devuelve al módulo de fusión de modalidades, de forma que la representación latente aprendida se ajuste a las tareas de predicción específicas. La función de pérdidas combinada queda como:

$$\begin{aligned} \mathcal{L} &= \alpha_1 \mathcal{L}_{rec} + \alpha_2 \mathcal{L}_{fit} + \mathcal{L}_{error} \\ &= \alpha_1 \sum_{v=1}^V \sum_{i=1}^N \sum_{t=1}^T \left\| o_{i,t}^{(v)} \left(f_v(\mathbf{h}_{i,t}; \Theta_v) - \mathbf{x}_{i,t}^{(v)} \right) \right\|_2^2 \\ &\quad + \alpha_2 \sum_{i=1}^N \sum_{t=1}^{T-1} |\mathbf{p}_{i,t} \odot (g(\mathbf{s}_{i,t}; \Omega) - \mathbf{s}_{i,t+1})| + \sum_{i=1}^N \sum_{t=T+1}^{T+F} |\mathbf{q}_{i,t} \odot (\mathcal{F}(\tilde{\mathbf{s}}_{i,t}) - \mathbf{y}_{i,t})| \end{aligned}$$

Donde α_1 y α_2 son hiperparámetros.

2.10.2. Resultados

Los datos utilizados vienen de la base de datos ADNI. El objetivo consiste en reconstruir las trayectorias de marcadores cognitivos (MMSE, ADAS-Cog, CDR-Global, CDRSB) basándose en las observaciones longitudinales de las modalidades PET, MRI y datos demográficos.

Los datos se alinean en intervalos de 6 meses. Las visitas de entrenamiento son toda hasta la última con valores disponibles de PET o MRI. Los sujetos que no tengan marcadores cognitivos más allá del inicio del estudio se eliminan. Para los que no los tengan más allá de la última visita con valores de PET o MRI, se recorta la secuencia, prediciendo la visita siguiente a la indicada. Las métricas utilizadas son el MAE y el coeficiente de correlación ponderado (wR). Además, se utiliza un proceso de validación cruzada con 10 particiones.

Los modelos con los que se comparan los resultados son SVR (*Support Vector Regressor*, Lasso, GRU) con variantes *Mean Filling* (MeanF), *Forward Filling* (FF), *Linear Filling* (LF) y *Model Filling* (MF) de imputación de datos. También se compara con otros modelos del estado del arte: *Convex Fused Sparse Group Lasso* (cFSGL), *Peephole LSTM* (PLSTM), *MinimalRNN*, *Temporal LSTM* (LSTM-T). Los hiperparámetros óptimos se hallan con *Hyperopt toolbox*, en el espacio de búsqueda de la tabla 2.8. La significancia estadística sobre MAE y wR de los resultados del modelo propuesto se comprueba con la prueba t.

Modelo	Hiperparámetro	Rango
Variantes Lasso	λ	$(10^{-3} - 10^3)$
Variantes SVR	Kernel	RBF
	C	$(10^{-2} - 10^2)$
	γ	$(10^{-2} - 10^0)$
	ϵ	$(10^{-2} - 10^2)$
Variantes GRU, MinimalRNN, Modelo propuesto	Número de parámetros ocultos (d_h)	{64, 128, 256}
	Tasa de aprendizaje	$(10^{-3} - 10^{-1})$
	Disminución de los pesos	$(10^{-3} - 10^{-1})$
	α_1 (si hay)	$(10^{-2} - 10^0)$
PLSTM, LSTM-T	α_2 (si hay)	$(10^{-1} - 10^1)$
	Número de parámetros ocultos (d_h)	{16, 32, 48, 64, 80, 96}
	Tasa de aprendizaje	$\{5 \times 10^{-5}, 5 \times 10^{-4}, 5 \times 10^{-3}, 5 \times 10^{-2}\}$
	Disminución de los pesos	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$
	α_2	$(10^{-2} - 10^0)$

Tabla 2.8: Espacio de búsqueda de hiperparámetros para los distintos métodos utilizados. Fuente: Xu et al.[50]

El modelo propuesto presenta unos resultados de MAE y wR significativamente mejores que los demás métodos. El MAE se mantiene estable en todos los puntos predichos, mientras que el wR disminuye en los últimos 3 puntos más alejados debido a los pocos datos disponibles.

Por otro lado, se realizaron estudios de ablación:

- Efecto de los datos de distintas modalidades: al entrenar el modelo únicamente con datos de una modalidad, el desempeño es peor que el modelo entrenado con múltiples modalidades.
- Efecto del módulo de fusión de modalidades: al concatenar las modalidades en lugar de obtener la representación latente, los resultados descienden significativamente.
- Efecto del aprendizaje colaborativo: al entrenar los módulos por separado se observa que el resultado es incluso peor que al eliminar el módulo de fusión de modalidades.
- Importancia de las distintas modalidades: el cambio más evidente es la eliminación de los datos demográficos, siendo el desempeño incluso mejor que usando todas las modalidades.

2.11. Aportaciones al trabajo

2.11.1. NIA-AA Research Framework: Toward a biological definition of Alzheimer's disease

Proporciona un marco de trabajo sobre el que comparar los resultados con otras investigaciones. Considera la enfermedad de Alzheimer como un continuo en lugar de una sucesión de etapas definidas por un diagnóstico. Define la enfermedad según el perfil AT(N).

2.11.2. Detectable Neuropsychological Differences in Early Preclinical Alzheimer's Disease: A Meta-Analysis

Tras realizar un meta-análisis para determinar el poder discriminativo de 7 dominios cognitivos, concluye que los *tests* neuropsicológicos son capaces de discernir las fases del AD preclínico. Por lo tanto, estos marcadores son los que se usarán en el estudio.

2.11.3. Estimating long-term multivariate progression from short-term data

Propone el modelo GRACE para obtener las trayectorias de corto y largo plazo de los marcadores del AD, así como la ordenación temporal de los sujetos. Es uno de los métodos paramétricos empleados.

2.11.4. Learning Multimodal Digital Models of Disease Progression from Longitudinal Data: Methods & Algorithms for the Description, Prediction and Simulation of Alzheimer's Disease Progression

Describe el fundamento matemático de Leaspy. Se trata del otro método paramétrico utilizado.

2.11.5. Robust parametric modeling of Alzheimer's disease progression

Describe el fundamento matemático de RPDPM. Aunque no se ha entrenado en este estudio, se han obtenido sus resultados para compararlos con Leaspy y GRACE. Por otro lado, propone la estimación del diagnóstico según un clasificador de Bayes.

2.11.6. Temporal modeling and AT profiles in the early phase of Alzheimer's disease

Aplica el modelo GRACE para capturar la progresión de la enfermedad. Discierne entre sujetos A^+ y A^- . Propone un método para entrenar y evaluar los DPM para diversas combinaciones de marcadores. Este estudio sigue una estructura similar al buscar los vectores de características óptimos y definir métricas de evaluación de los modelos.

2.11.7. Predicting Alzheimer's disease progression using deep recurrent neural networks

Describe el funcionamiento de *minimalRNN*. Propone las estrategias de *Forward Filling*, *Linear Filling* y *Model Filling* para tratar con valores faltantes. Sirve como referencia en el entrenamiento de las redes neuronales recurrentes, aunque no se utilice *minimalRNN*. También se aprovecha la idea del *Linear Filling* en la imputación de valores faltantes.

2.11.8. CARRNN: A Continuous Autoregressive Recurrent Neural Network for Deep Representation Learning From Sporadic Temporal Data

Describe el fundamento matemático de CARRNN. Propone el *time binning* para agrupar las observaciones disponibles en intervalos temporales. Aunque CARRNN no se utilice en este estudio, el *time binning* es la estrategia utilizada en la alineación temporal.

2.11.9. Deep recurrent model for individualized prediction of Alzheimer's disease progression

Describe el funcionamiento de una LSTM basada en 3 módulos: imputación, estimación y predicción. Propone el entrenamiento colaborativo de los mismos en base a una función de pérdidas global. Además, incorpora dependencias temporales y multivariantes, así como máscaras con las posiciones de los valores faltantes y las diferencias temporales entre observaciones sucesivas. El modelo desarrollado es una versión inicial de esta propuesta.

2.11.10. Multi-modal sequence learning for Alzheimer's disease progression prediction with incomplete variable-length longitudinal

Describe el funcionamiento de una LSTM que incorpora módulos de fusión de modalidades para analizar la evolución desde distintos puntos de vista. Propone el entrenamiento colaborativo de la red junto con estos módulos en base a una función de pérdidas global. El modelo desarrollado es una versión inicial de esta propuesta.

Capítulo 3

Materiales

Los datos utilizados proceden de la base de datos de ADNI (*Alzheimer's Disease Neuroimaging Initiative*). Los marcadores pueden dividirse en neuropsicológicos (NM), líquido cefalorraquídeo (CSF), resonancia magnética (T1-MRI), tomografía por emisión de positrones (PET) y genéticos.

Además de los lenguajes de programación en los que se entrenan los modelos (R, Python y MATLAB), se describen otras herramientas usadas. Cabe destacar, el manejo y creación de entornos en Magerit-3, el sistema de cómputo científico del CesViMa.

3.1. Marcadores ADNI

3.1.1. NM

Los test neuropsicológicos (NM) son pruebas que examinan el desempeño cognitivo en áreas como la memoria, aprendizaje, lenguaje, orientación espacio-temporal, nivel de atención ... A diferencia de otras pruebas (p.ej. T1-MRI), son más rápidos, sencillos y económicos de realizar, ya que no requieren maquinaria especializada. Los principales son:

- ADAS (*Alzheimer's Disease Assessment Scale*)

Son pruebas que evalúan los dominios cognitivos y no cognitivos en personas con AD. Especial importancia tiene la subescala ADAS-Cog, que se centra en los aspectos cognitivos. Mide principalmente el lenguaje y la memoria, y tarda en realizarse aproximadamente 30 minutos. Es más completa que la MMSE. Hay varias versiones:

- ADAS11: 11 tareas.
- ADAS13: 13 tareas. Incluye las 11 tareas de ADAS11 y 2 adicionales: funciones ejecutivas y actividades visuoespaciales.
- ADASQ4: 4 tareas. Es una versión simplificada de la prueba ADAS-Cog y más rápida de realizar.

- MMSE (*Mini Mental State Examination*)

Es un test que se realiza entre 5 y 10 minutos. Permite obtener una estimación inicial o la evolución del estado cognitivo del paciente. Se puntúa de 0 a 30 (aunque existen distintas versiones). Se evalúan la orientación temporal y espacial, memoria inmediata y retención, concentración y memoria de trabajo, lenguaje, y praxis constructiva gráfica (p.ej. copiar un dibujo).

- RAVLT (*Rey Auditory Verbal Learning Test*)

Evalúa la memoria verbal auditiva y el aprendizaje. Se desarrolla de la siguiente forma:

1. Se lee en voz alta al participante 15 palabras en voz alta y se le pide que las recuerde inmediatamente.
2. Se presenta una nueva lista con otras 15 palabras y se le pide al participante que las recuerde inmediatamente.
3. El participante debe recordar las palabras de la lista original después de la introducción de la última lista.
4. Después de 20-30 minutos de realización de otras tareas, el participante deberá volver a recordar las palabras de la lista original.
5. Se le entrega al participante una lista de 50 palabras con palabras de ambas listas. Este deberá ser capaz de reconocer las palabras de la lista original.

- FAQ (*Functional Assessment Questionnaire*)

Es una herramienta que evalúa el estado funcional de personas con demencia: habilidades físicas (p.ej. vestirse), del lenguaje, movilidad, alimentación.

- Ecog (*Everyday Cognition*)

Evalúa aspectos cognitivos de la vida diaria relativos a la memoria (p.ej. productos de la compra) y el lenguaje (p.ej. explicar un programa de TV). Los resultados pueden ser:

- Mejor o sin cambios
- Ocasionalmente peor
- Habitualmente un poco peor
- Habitualmente mucho peor

- Otros:

- CDR (*Clinical Dementia Rating*)
- MoCA (Montreal Cognitive Assessment)
- LM (*Logical Memory*)
- LDELTOTAL (*Long Delay Free Recall Total*)
- DSST (*Digit Symbol Substitution Test*): DIGITSCOR en ADNI.
- ADCS-PACC (*Alzheimer's Disease Cooperative Study-Preclinical Alzheimer Cognitive Composite*)
- TMT (*Trail Making Test*): TRABSCOR en ADNI.

3.1.2. CSF

El líquido cefalorraquídeo (CSF) se encuentra en el sistema nervioso central (CNS) y se encarga de proteger frente a impactos a estructuras como el cerebro o la médula espinal. También elimina los desechos producidos por el cerebro y ayuda al correcto funcionamiento del CNS. Las pruebas de CSF consisten en extraer una muestra de este líquido mediante punción lumbar para analizar biomarcadores como:

- $A\beta$: beta amiloide. En la AD se observan depósitos de $A\beta$ (placas seniles) extracelulares.
- Tau: proteína Tau.
- pTau: proteína Tau fosforilada. En la AD, esta proteína está anormalmente fosforilada y forma ovillos neurofibrilares en el interior de las células.

También es útil analizar los ratios:

- $\text{Tau}/A\beta$
- $\text{pTau}/A\beta$

3.1.3. T1-MRI

Son medidas de espesores corticales y volúmenes cerebrales obtenidas por imágenes de resonancia magnética.

Marcadores de volumetría

Miden el volumen normalizado de distintas áreas del cerebro. La normalización se realiza dividiendo la medida obtenida entre el volumen intracraneal del individuo:

- NHV: volumen hipocampal normalizado.
- NAV: volumen de la amígdala cerebral normalizado.
- NPV: volumen del putamen normalizado.
- NCV: volumen del caudete normalizado.
- NPaV: volumen del globo pálido normalizado.
- NVEN: volumen del ventrículo lateral normalizado
- NEV: volumen de la corteza entorrinal normalizado.
- NVV: volumen ventricular normalizado.
- NFV: volumen del giro fusiforme normalizado.
- NMV: volumen del lóbulo temporal medial normalizado.

Marcadores de espesores corticales

Miden el grosor de la corteza cerebral con el fin de obtener información sobre su estructura y organización.

- ECT: espesor de la corteza entorrinal.
- PCT: espesor de la corteza cingulada posterior.
- IPT: espesor del lóbulo parietal inferior.
- MTT: espesor del lóbulo temporal medial.
- ITT: espesor del lóbulo temporal inferior.
- SPT: espesor del lóbulo parietal superior.

- TPT: espesor del polo temporal.
- PRT: espesor del precúneo.
- MeT: suma de los espesores.

3.1.4. PET

La tomografía por emisión de positrones (PET) es una prueba de imagen que evalúa el funcionamiento de tejidos y órganos. Un marcador radiactivo es inyectado por vía intravenosa en una zona con altos niveles de actividad química. Dicho marcador puede medir procesos como cambios en el metabolismo, flujo sanguíneo o uso de oxígeno. Los principales marcadores son:

- FDG: fluorodeoxiglucosa. Para medir los niveles de $A\beta$.
- AV-1451 (AV-45): florbetapir. Para medir los niveles de Tau.

3.1.5. Marcadores genéticos

La *APOE* $\epsilon 4$ es un alelo de la apolipoproteína E. Se considera el principal factor de riesgo genético para la AD. En la población ADNI se denota con un 1 si el sujeto es portador y con un 0 si no lo es.

3.2. Programas utilizados

A continuación, se describen los programas utilizados en el estudio. El documento ha sido redactado en \LaTeX .

3.2.1. PuTTY

PuTTY es un cliente SSH, Telnet, rlogin y TCP raw con licencia libre. Es gratuito. Está disponible Windows, aunque se está portando a sistemas Unix y macOS. Se ha utilizado para conectarse a máquinas remotas que ofrecen una capacidad computacional y de almacenamiento elevada.

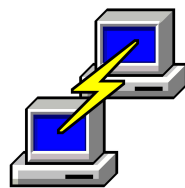


Figura 3.1: Logo de PuTTY [12]

3.2.2. Xming

Xming es un software de código abierto que proporciona una forma de ejecutar aplicaciones gráficas de forma remota en plataformas Windows. Es gratuito. Utiliza el sistema de ventanas X Window. Se ha utilizado para disponer de una interfaz gráfica para las aplicaciones ejecutadas en máquinas remotas Unix (p.ej. MATLAB). Para ello, es necesario activar en PuTTY la casilla *Enable X11 forwarding*.



Figura 3.2: Logo de Xming [23]

3.2.3. FileZilla

FileZilla es una aplicación FTP libre, gratuita y de código abierto que soporta los protocolos FTP, SFTP y FTPS. Fue diseñada para Windows, pero está disponible también para otros sistemas operativos como macOS y GNU/Linux. Se ha utilizado para la transferencia de archivos con las máquinas remotas.



Figura 3.3: Logo de FileZilla [4]

3.2.4. MATLAB

MATLAB es un sistema de cómputo numérico que ofrece IDE con un lenguaje de programación propio (M). Es privativo y está disponible para Unix, Windows, macOS y GNU/Linux. Entre sus características básicas, cabe destacar la manipulación de matrices, representación de datos y funciones, implementación de algoritmos y comunicación con otros lenguajes de programación. Se ha usado la versión R2020b.

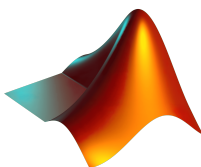


Figura 3.4: Logo de MATLAB [8]

3.2.5. R

R es un entorno y lenguaje de programación libre y de código abierto enfocado en la estadística. Está disponible para Windows, macOS, Unix y GNU/Linux. Es muy popular en campos como machine learning, minería de datos, bioinformática ... Como IDE se ha usado RStudio [16].

3.2.6. Python

Python es un lenguaje de programación multiparadigma, multiplataforma e interpretado, cuya filosofía se centra en la legibilidad del código. Soporta programación orientada a objetos y posee una licencia de código abierto. Como IDE se ha usado Spyder [18].



Figura 3.5: Logos de R (figura 3.5a) [14] y RStudio (figura 3.5b) [16].

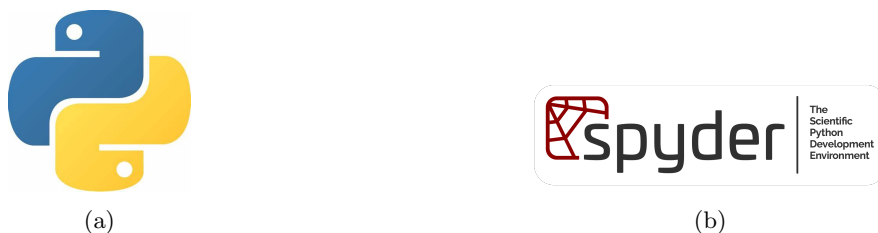


Figura 3.6: Logos de Python (figura 3.9) [13] y Spyder (figura 3.6b) [18].

3.2.7. Anaconda

Anaconda es una distribución que aloja los paquetes esenciales de Python, herramientas como Jupyter y RStudio, y un gestor de paquetes (Conda). Esto simplifica la gestión de paquetes y dependencias, así como el despliegue de proyectos de ciencia de datos, aprendizaje automático y computación científica. También ofrece herramientas de seguridad, como el *Package Security Manager*, que ayuda a garantizar la integridad de los entornos de *software*. Además, cuenta con soluciones empresariales con funciones para el trabajo colaborativo, seguridad y escalabilidad, entre otras.

Se ha usado para crear entornos de trabajo con las versiones correctas de los paquetes y sus dependencias de Python y R.



Figura 3.7: Logo de Anaconda [2]

3.2.8. tmux

Tmux es un multiplexor de terminal: permite crear, acceder y controlar varios terminales desde una única pantalla. Además, puede separarse de la pantalla y seguir ejecutándose en segundo plano, para volver a juntarse posteriormente. Funciona en OpenBSD, FreeBSD, NetBSD, Linux, macOS y Solaris. Se ha utilizado para ejecutar tareas pesadas que deben seguir corriendo de forma ininterrumpida, incluso después de desconectar la sesión SSH.

Los comandos básicos para su operación se resumen a continuación:

- `tmux new`: inicia una nueva sesión.



Figura 3.8: Logo de tmux [21]

- `tmux rename-session -t <nombre actual ><nombre nuevo >`: cambia el nombre de una sesión.
- `tmux attach-session -t <nombre de sesión >`: adjunta el terminal actual a una sesión existente.
- `tmux ls`: lista todas las sesiones.
- `tmux detach`: separa una sesión, pasando a ejecutarla en segundo plano.
- `tmux kill-session -t <nombre de sesión >`: finaliza una sesión.
- `tmux find-window -t <nombre de sesión ><texto >`: busca un texto específico en todas las ventanas de una sesión.
- `tmux send-keys -t <nombre de sesión ><comando >`: envía una señal a todas las ventanas de una sesión.
- `tmux rename-window -t <sesión:ventana ><nombre nuevo >`: cambia el nombre de una ventana.
- `Ctrl+b c`: crea una nueva ventana dentro de la sesión actual.
- `Ctrl+b n`, `Ctrl+b p`: cambia a la ventana siguiente o anterior, respectivamente.
- `Ctrl+b <número de ventana >`: cambia a la ventana indicada.
- `Ctrl+b %`, `Ctrl+b "`: divide la ventana actual en 2 paneles verticales u horizontales, respectivamente.
- `Ctrl+b [flechas]`: cambia al panel al que apunta la flecha.
- `Ctrl+b x`: elimina un panel.
- `Ctrl+b [, Ctrl+b]`: modo copiar y pegar texto, respectivamente.
- `exit`: cierra el panel actual.

3.2.9. Magerit-3

Magerit-3 es un sistema de cómputo científico destinado a la ejecución de cargas de trabajo científico que requieren cómputo intensivo.

Es un clúster de propósito general compuesto por 72 nodos ThinkSystem SD530, cada uno de ellos equipado con procesadores Intel Xeon Gold 6230 y 192 GiB de RAM, y 48 nodos ThinkSystem SD530, cada uno de ellos equipado con procesadores Intel Xeon Gold 6240R y 768 GiB de RAM. Esta configuración es capaz de proporcionar una potencia pico de 370.49 TFLOPS (DP). Adicionalmente, se dispone de nodos ThinkSystem SR670 con aceleradores (GPU) específicos.

Cada actividad tiene asignado 1 TB de almacenamiento compartido por todos los miembros y una carpeta de proyecto con el formato `/home/<code >/` para almacenar la

información.

Magerit se explota mediante trabajos *batch* usando SLURM como gestor y planificador de recursos. Para ejecutar se indican las características del trabajo que se necesita y el sistema se encargará de reservar los recursos necesarios y ejecutar las tareas.

El trabajo se envía al sistema ejecutando `sbatch job.sh` y ejecutaría 1 tarea (1 CPU) durante un máximo de 24 horas. Estos parámetros se pueden modificar mediante directivas de SLURM.



Figura 3.9: Logo de CesViMa [3]

En el fichero `.sh` se añaden las directivas SLURM, que configuran cómo se ejecuta el trabajo. A continuación, se incluyen algunas [17]:

- `SBATCH --job-name=<name>`: nombre del trabajo.
- `SBATCH --nodes=<#>`: número de nodos a utilizar.
- `SBATCH --ntasks=<# tags>`: número de tareas a ejecutar.
- `SBATCH --mem-per-cpu=<#>`: memoria RAM (MiB) requerida por CPU.
- `SBATCH --time=[[DD-]HH:]MM:SS`: duración máxima de ejecución del trabajo.
- `SBATCH --mail-type=ALL`: enviar correos electrónicos al usuario en todos los eventos de trabajo.
- `SBATCH --mail-user=email`: dirección de correo para el envío de notificaciones de eventos.
- `SBATCH --output=out-%j.log`: fichero para almacenar la salida estándar del trabajo.
- `SBATCH --error=err-%j.log`: fichero para almacenar la salida de errores del trabajo.
- `SBATCH --chdir=<dirname>`: directorio de trabajo.

Serán útiles los siguientes mandatos SLURM para controlar los trabajos en ejecución:

- `sbatch`: envía un trabajo al sistema devolviendo su identificador.
- `squeue`: lista los trabajos encolados en el sistema (aquellos que están ejecutando o esperando para ejecutar).

- `scancel`: cancela un trabajo encolado. Si el trabajo ya estaba ejecutando se abortará la ejecución en ese momento.
- `sstat`: muestra estadísticas de uso de un trabajo que esté ejecutando en ese momento.
- `sacct`: muestra información de la ejecución de un trabajo ya finalizado.

GRACE

La ejecución de GRACE requiere instalar algunas librerías en el sistema. Los pasos a seguir se describen a continuación:

1. Iniciar sesión en Magerit-3 a través de PuTTY.
2. Ejecutar los siguientes mandatos (carga y ejecución de R):
 - a) `module load R`
 - b) `R`
3. Usar el mandato de R `install.packages(#nombre_libreria)` para la instalación de las librerías previas requeridas:
 - `devtools`
 - `mvtnorm`
 - `readxl`
4. Ejecutar los siguientes mandatos en R para instalar GRACE:
 - a) `install_bitbucket("mdonohue/grace", lib=~ /R/libs")`
 - b) `install.packages("fda", lib=~ /R/libs")`
 - c) `install.packages("writexl", lib=~ /R/libs")`
 - d) `.libPaths(c(.libPaths(), "~ /R/libs/"))`

Leaspy

Leaspy requiere un entorno específico de ejecución con determinadas versiones de las librerías. Conda permite la configuración de dicho entorno. Los pasos a seguir son los siguientes:

1. Iniciar sesión en Magerit-3 a través de PuTTY.
2. Ejecutar los siguientes mandatos:
 - a) `source ~/.bashrc`
Ejecuta el contenido del archivo `.bashrc`, que inicia una nueva *shell* con las configuraciones específicas del usuario.
 - b) `conda create --name leaspy python=3.8`
Crea un nuevo entorno conda, `leaspy`, en el que se instalará la versión 3.8 de Python.
 - c) `conda activate leaspy`
Activa el entorno `leaspy`.

- d) `pip install leaspy`
Instala el paquete `leaspy`.
- e) `conda install numpy=1.22.4 pandas=1.5.3 openpyxl=3.1.2 ...`
Instala las versiones de las librerías necesarias para la ejecución de los trabajos, por ejemplo `numpy`, `pandas`, y `openpyxl`.
- f) `conda install -c conda-forge mpi4py`
Instala `mpi4py` a través del canal `conda-forge`. Es necesario especificar este canal debido a que incluye las dependencias requeridas para que el paquete funcione correctamente.

Redes neuronales recurrentes

La construcción de redes neuronales recurrentes en Python requieren un entorno Conda en el que esté instalado *Tensorflow*. A continuación, se exponen los pasos a seguir para la instalación del entorno:

1. Iniciar sesión en Magerit-3 a través de PuTTY.
2. Ejecutar los siguientes mandatos:
 - a) `source ~/.bashrc`
 - b) `conda create --name rnn python=3.11.5`
 - c) `conda activate rnn`
 - d) `conda install numpy=1.24.3 pandas=2.2.3 openpyxl=3.1.5 statsmodels=0.14.0
scipy=1.11.1 scikit-learn=1.3.0`
 - e) `conda install -c conda-forge mpi4py`
 - f) `conda install pytorch torchvision torchaudio -c pytorch`
 - g) `mkdir /home/usuario/pip_cache`
 - h) `export TMPDIR=/home/usuario/pip_cache`
 - i) `pip install tensorflow==2.16.1`

Las últimas 3 líneas permiten instalar *Tensorflow* correctamente en Magerit-3. Podría pensarse que con ejecutar `pip install tensorflow==2.16.1` es suficiente; sin embargo, ocurre el siguiente error:

ERROR: Could not install packages due to an OSError: [Errno 28] No queda espacio en el dispositivo

En algunos entornos de trabajo se limita el espacio del directorio temporal `/tmp`, por lo que paquetes pesados como *Tensorflow* requieren más espacio del disponible. La solución consiste en crear un nuevo directorio, `/home/usuario/pip_cache`, en una partición diferente en la que se disponga de más espacio para la descarga. Seguidamente, se apunta la variable de entorno `TMPDIR` al directorio creado y se procede con la instalación.

3.2.10. LaTeX

\LaTeX es un sistema de composición de textos de alta calidad que incluye funciones para la producción de documentación técnica y científica. Es gratuito y de código libre. Se ha utilizado en la redacción del presente documento.

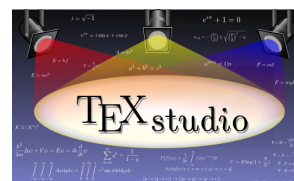
Se ha usado TeXstudio como IDE, con la distribución MiKTeX y el compilador PdfLaTeX¹. La bibliografía se ha generado con BibTeX.



(a)



(b)



(c)

Figura 3.10: Logos de \LaTeX (figura 3.10a) [7], MiKTeX (figura 3.10b) [11] y TeXstudio (figura 3.10c) [20].

¹La lista de acrónimos se genera con el comando `makeglossaries`, pasando el nombre del documento como argumento, desde la terminal de Windows. Es preciso haber compilado previamente el documento con PdfLaTeX. Tras ejecutar el comando, debe volver a compilarse el documento. Se requiere tener *Perl* instalado para poder ejecutar `makeglossaries`. Puede descargarse *Strawberry Perl* para Windows en [19].

Capítulo 4

Métodos

El procedimiento seguido se estructura en varias etapas. Comienza con la descarga de los datos de ADNI y su preprocesamiento con el fin de eliminar inconsistencias y caracteres extraños. A continuación, se define la población de estudio, clasificándola en diversos grupos según su diagnóstico y se realiza un análisis inicial para caracterizarla. Posteriormente, se explican las estrategias de entrenamiento, personalización y predicción de los modelos paramétricos y las redes neuronales recurrentes. Una vez definidos los algoritmos, se realiza un análisis masivo para hallar los marcadores que mejor capturan la enfermedad. Finalmente, las combinaciones óptimas se someten a pruebas de robustez ante valores faltantes, análisis en función del número de visitas, pronóstico y reconstrucción de la historia natural que permiten comparar los algoritmos entre sí. Adicionalmente, se incluye una sección en la que se detalla el proceso de paralelización de tareas en Magerit-3 mediante MPI.

4.1. Obtención de los datos

Los datos se obtienen de la página de ADNI [1]. Una vez en la página, acceder a **Download ->Study Data**, buscar los formatos *.csv* de ADNIMERGE y descargarlos (Figura 4.1).

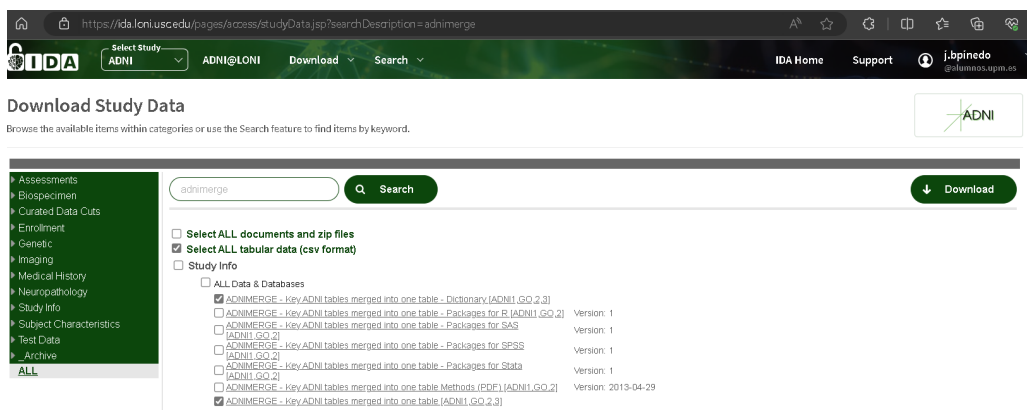


Figura 4.1: Descarga de datos de la población ADNIMERGE

4.2. Preprocesamiento de los datos

Los datos descargados de ADNIMERGE presentan ciertos problemas:

- Están en formato *.csv*.
- Los valores de algunos biomarcadores presentan saturaciones (p.ej. ">1700" no se trata como valor numérico, sino como string).
- En algunos casos, se usa "." como separador decimal en vez de ",",.
- Los datos vienen desordenados.

El archivo `/demo/clasificacion_poblacion/getADNIMERGE.ipynb` convierte el archivo a *.xlsx*, elimina el problema de las saturaciones y de los separadores decimales. Además, ordena los datos por RID y VISCODE.

Algoritmo 1 getADNIMERGE

Entrada:

- Tabla ADNIMERGE raw en formato *.csv*

Salida:

- Tabla ADNIMERGE procesada en formato *.xlsx*

- 1: Leer los datos del fichero *.csv* descargado de ADNIMERGE
 - 2: Convertir la fecha a formato estándar
 - 3: Ordenar las filas por RID y VISCODE
 - 4: **for** dato en la base de datos **do**
 - 5: Elimina caracteres especiales
 - 6: **end for**
 - 7: Convertir y guardar la información en formato *.xlsx*
-

4.3. Población de estudio

El estudio es sobre la fase temprana de la enfermedad: se analiza la evolución de pacientes cognitivamente normales hacia deterioro cognitivo leve. Por lo tanto, se tendrán en cuenta los siguientes grupos:

- Sujetos sCU (*Stable Cognitively Unimpaired*): sujetos cognitivamente normales en la primera y última visita.
- Sujetos pCU (*Progressive Cognitively Unimpaired*): sujetos inicialmente cognitivamente normales que durante el estudio convierten a deterioro cognitivo leve o demencia.
- Sujetos sSMC (*Stable Significant Memory Concern*): sujetos que comienzan el estudio con queja subjetiva y no convierten a deterioro cognitivo leve o demencia.
- Sujetos pSMC (*Progressive Significant Memory Concern*): sujetos que comienzan el estudio con queja subjetiva y convierten a deterioro cognitivo leve o demencia.
- Sujetos sMCI (*Stable Mild Cognitive Impairment*): sujetos con deterioro cognitivo leve en la primera y última visita.

4.4. Determinación del tiempo de conversión clínico

La posición relativa al tiempo cero de cada paciente determina la evolución de su enfermedad. Esta información es necesaria para la construcción de modelos, por lo que su

cálculo debe ser lo más exacto posible.

Hasta ahora, este parámetro se había calculado usando el criterio del VISCODE, es decir, el mes de visita (la separación temporal mínima entre 2 visitas es de 6 meses con dicho criterio). Usando el criterio del EXAMDATE logra mayor precisión, ya que se coge la fecha en días como referencia. Aplicando este nuevo criterio:

- Los sujetos sCU tendrán tiempos de conversión negativos en todas sus visitas. Esto se corresponde con que todavía no habrán alcanzado el deterioro cognitivo leve. El tiempo de conversión de su última se anota como 0 (aunque en realidad sea desconocido, resulta útil para tener una referencia).
- Los sujetos MCI tendrán tiempos de conversión positivos, debido a que en su primera visita su diagnóstico ya era MCI. Sin embargo, se anotará su primera visita como tiempo de conversión 0 por la misma razón que en los sCU (no se conoce cuándo realmente convirtieron, por lo que se considera su primera visita como dicho evento).
- Los sujetos pCU tendrán tiempos de conversión negativos en sus primeras visitas y positivos en las últimas. Estos son los sujetos más interesantes, ya que se tiene el tiempo de onset real (el 0 estará entre la primera y la última visita) en vez de una estimación, a diferencia de los casos anteriores. En este caso, además, se cogerá como tiempo de onset la media entre la primera visita en la que el paciente cambia de diagnóstico y la visita anterior. Esto es porque cuando cambia de diagnóstico ya es MCI, pero en la visita anterior era CU, por lo que el tiempo cero se situará entre medias de ambas fechas.

El algoritmo 2 calcula el tiempo de conversión de los pacientes:

Algoritmo 2 tonset

Entrada:

- Tabla ADNIMERGE procesada

Salida:

- Tabla ADNIMERGE con columna de tiempo de conversión en días

```

1: Leer tabla ADNIMERGE .xlsx
2: Convertir la columna EXAMDATE a días desde el baseline
3: for sujeto en tabla do
4:   if sujeto es sCU then
5:     Establecer última visita como onset
6:   else if sujeto es MCI then
7:     Establecer visita inicial como onset
8:   else if sujeto es pCU o pSMC then
9:     Buscar la primera visita en la que se diagnostica MCI
10:    Calcular la media en días entre el tiempo hasta la visita de cambio de diagnóstico y la anterior
11:   end if
12:   Calcular la diferencia entre el tiempo de onset y cada una de las visitas
13: end for

```

4.5. Caracterización de la población de estudio

La tabla ADNIMERGE contiene información de los biomarcadores de los pacientes en cada visita realizada. Una vez ordenada y limpiados los caracteres problemáticos, se

procede a clasificar y caracterizar la población.

La clasificación se llevará a cabo según varios criterios:

- Biomarcadores: si el sujeto ha sido sometido solamente a test neuropsicológicos será NM. Si, además, presenta datos del CSF pertenecerá al grupo NM + CSF.
- Diagnóstico clínico: CU, SMC, MCI, AD.
- Estable (los diagnósticos inicial y final coinciden) o progresivo (los diagnósticos inicial y final no coinciden).

Para cada grupo, se calcula la media, desviación típica, niveles máximo y mínimo de cada biomarcador. Se realizan test ANOVA y Chi Cuadrado con prueba post-hoc (método de Tukey) para observar entre qué grupos existen diferencias significativas en los resultados.

Los scripts `categories_nm_csf.ipynb`, `categories_nm.ipynb`, `convert_censure_nm_csf.ipynb` y `convert_censure_nm.ipynb` en la carpeta `/demo/clasificacion_poblacion` se encargan de esta caracterización.

Algoritmo 3 `categories_nm` y `categories_nm_csf`

Entrada:

- Tabla ADNIMERGE procesada en formato `.xlsx`

Salida:

- Tablas con la caracterización de la población en formato `.h5`
- Datos con los registros clasificados por grupos según el diagnóstico en formato `.h5`

```

1: Leer tabla ADNIMERGE .xlsx
2: for diagnóstico en [CU, SMC, MCI, AD] do
3:   Obtener media, desviación típica, mínimo y máximo de cada biomarcador
4: end for
5: for cada biomarcador analizado do
6:   if la medida es categórica then
7:     Aplicar prueba Chi Cuadrado
8:   else
9:     Aplicar prueba ANOVA
10:  end if
11: end for
12: Convertir y guardar las tablas resultantes en formato .h5
13: Convertir y guardar los datos filtrados por grupos en formato .h5

```

4.6. Métodos paramétricos de progresión de la enfermedad

Los DPM (*Disease Progression Modelling*) paramétricos estiman la historia natural de la enfermedad, el desplazamiento temporal y los parámetros característicos de cada sujeto. De esta forma, se pueden averiguar las trayectorias individuales de los sujetos observados y la trayectoria general para un sujeto genérico. Los métodos GRACE y Leaspy realizan dichas tareas.

El procedimiento empleado consta de varias etapas:

Algoritmo 4 convert_censure_nm y convert_censure_nm_csf

Entrada:

- Datos con los registros clasificados por grupos según el diagnóstico en formato `.h5`

Salida:

- Tablas con la caracterización de la población (convertores y no convertores) en formato `.h5`
- Datos con los registros clasificados en convertores y no convertores en formato `.h5`

```

1: Leer datos de la población clasificada según diagnóstico
2: for grupo en [CN, SMC] do
3:   for RID en grupo do
4:     if diagnóstico inicial == diagnóstico final then
5:       Clasificar al paciente como estable
6:       Calcular tiempo de censura del paciente
7:     else
8:       Clasificar al paciente como convertor
9:       Calcular tiempo de conversión del paciente
10:    end if
11:  end for
12: end for
13: Aplicar el algoritmo 3 para caracterizar los nuevos grupos y guardar los resultados

```

1. Preparación de los datos: se crea una tabla con los datos necesarios para entrenar cada algoritmo. Esto implica seleccionar los sujetos y marcadores de interés, y desechar el resto. Los valores de los marcadores se pasan a escala percentil para normalizarlos y evitar valores atípicos. Además, se consigue tener todas las variables en una escala común. Es importante que todos los marcadores evolucionen en el mismo sentido. Por ejemplo, una puntuación alta en ADAS13 indica un estado avanzado de la enfermedad; sin embargo, con RAVLT Learning sucede al revés. En este caso, habría que invertir las puntuaciones de uno de los marcadores (según el sentido que se quiera definir).
2. Entrenamiento de los modelos: con los datos del paso anterior, se entrenan los modelos y se realizan las estimaciones. Se usa el modelo multivariante.
3. Comparación de resultados: se comparan los resultados obtenidos por ambos modelos. Un alto grado de correlación podrá servir a modo de validación.

Cabe decir que los pacientes con algún marcador NaN o con una sola visita se eliminan antes de ser introducidos en Leaspy. GRACE puede encargarse de estos datos sin problema. Por lo tanto, el set de datos usado en Leaspy será un subconjunto de GRACE. Aquellas visitas cuyos marcadores sean todos NaN serán eliminadas debido a que no aportan información.

4.6.1. GRACE

Se introduce un set de datos de cuatro columnas:

- `id`: identificador del sujeto.
- `t`: tiempo a corto plazo de las observaciones a corto plazo. Se calcula en años centrales: $t_{ij}^c = t_{ij} - (t_{i1} + t_{iend})/2$.
- `y`: valor del biomarcador entre 0 y 1.

- Outcome: identificador de cada marcador.

Para introducir los datos en GRACE, habrá que apilar los biomarcadores de forma que se tenga una única columna, y, con todos sus valores.

El resultado devuelto por GRACE es un *dataframe* con los parámetros individuales de cada sujeto por marcador y las estimaciones de g_j (g_{hat}) y de Y_{ij} (y_{ihat}). Los valores de α_{0ij} y α_{1ij} representan el desplazamiento espacial para cada marcador, es decir, la desviación frente a la historia natural. El valor de γ_i es el desplazamiento espacial e indica en qué punto de la enfermedad se sitúa el sujeto.

Personalización de GRACE

Según se comentó en [28], se busca un modelo de la forma:

$$Y_{ij} = g_j(t + \gamma_i) + \alpha_{0ij} + \alpha_{1ij}t + \epsilon_{ij}(t)$$

En el *training* se estima g_j , correspondiente con la función que modela la evolución de cada biomarcador. También se determina γ_i , α_{0ij} y α_{1ij} para los sujetos de entrenamiento. Cada vez que se registra un nuevo sujeto en la base de datos, son desconocidos sus parámetros γ_i , α_{0ij} y α_{1ij} . Estos se calculan durante la fase de personalización. De esta forma, cada sujeto evolucionará según una trayectoria única individualizada.

A continuación, se describe el procedimiento:

1. **Calcular γ_{inew} :** se trata del valor de γ_i que le correspondería al nuevo sujeto por visita y marcador si su trayectoria correspondiera con g_j . Como los valores de g_j se han estimado durante el entrenamiento (g_{hat}), es posible determinar g_{hinw} (inversa de g_{hat}) mediante interpolación lineal con los valores de *train*. Se está calculando la inversa, por lo que los valores de g_{hat} se colocan en el eje de abscisas y los de $t + \gamma_i$ en el de ordenadas.

Introduciendo los valores reales del biomarcador en g_{hinw} , se tiene $t + \gamma_{inew}$. Por tanto, γ_{inew} se obtiene restando t al resultado anterior. Cabe resaltar que después de este paso se habrán obtenido tantos valores de γ_{inew} como el resultado del producto del número de biomarcadores por el número de visitas.

2. **Calcular γ_i :** idealmente, todos los valores obtenidos de γ_{inew} deberían coincidir. Sin embargo, la estimación no es perfecta y los γ_{inew} difieren entre sí. Por lo tanto, se adopta una solución de compromiso al calcular γ'_i como la media de los γ_{inew} (para todas las visitas y marcadores).

El valor obtenido de γ'_i no tiene ningún significado por sí mismo: es necesaria una referencia. En consecuencia, se resta la media de los desplazamientos temporales de todos los sujetos de entrenamiento a γ'_i , obteniendo γ_i . Valores de γ_i mayores de 0 indicarán que el sujeto está más avanzado en la enfermedad que la media. Valores por debajo de 0 indicarán lo contrario.

3. **Calcular g_{hat} en $t + \gamma_i$:** γ_i se ha estimado en el paso anterior y g_{hat} se halla en el entrenamiento, por lo que el cálculo de $g_{hat}(t + \gamma_i)$ es directo.
4. **Calcular α_{0ij} y α_{1ij} :** determinan la traslación espacial y buscan minimizar el error cometido en $Y_{ij} - g_j(t + \gamma_i)$. En el entrenamiento se usó un modelo lineal de efectos

mixtos (LME). Este procedimiento no es aplicable en la personalización porque implicaría reentrenar el LME. Por lo tanto, se estiman α_{0ij} y α_{1ij} por otras vías. Pueden darse 3 situaciones en función del número de observaciones de cada marcador:

- No hay observaciones: tanto α_{0ij} como α_{1ij} valen 0.
- Hay una única observación: α_{0ij} corresponde a la diferencia $Y_{ij} - g_j(t + \gamma_i)$, mientras que $\alpha_{1ij} = 0$.
- Hay más de una observación: se aproxima la diferencia $Y_{ij} - g_j(t + \gamma_i)$ por mínimos cuadrados para minimizar el error.

En el algoritmo 5 se muestra el pseudo-código del procedimiento descrito.

Algoritmo 5 Personalización de GRACE

Entrada:

- Datos del/los nuevo(s) sujeto(s)
- Vector de características
- g_{hat}
- g_{hinv}
- Valor medio de γ_i para los sujetos de entrenamiento (γ_{imean})

Salida:

- γ_i
- α_{0ij}
- α_{1ij}

```

1: for marcador en marcadores do
2:   for visita en visitas do
3:     Calcular  $\gamma_{inew}$ 
4:   end for
5: end for
6:  $\gamma_i = \overline{\gamma_{inew}} - \gamma_{imean}$ 
7: for marcador en marcadores do
8:   for visita en visitas do
9:     Calcular  $g_{hat}(t + \gamma_i)$ 
10:  end for
11: end for
12: for marcador en marcadores do
13:   if número de visitas == 0 then
14:      $\alpha_{0ij} = 0$ 
15:      $\alpha_{1ij} = 0$ 
16:   else if número de visitas == 1 then
17:      $\alpha_{0ij} = Y_{ij} - g_j(t + \gamma_i)$ 
18:      $\alpha_{1ij} = 0$ 
19:   else State Calcular  $\alpha_{0ij}$  y  $\alpha_{1ij}$  por mínimos cuadrados
20:   end if
21: end for

```

La comprobación de los resultados de la personalización se realiza de varias formas:

- Comparación con la función de entrenamiento: consiste en entrenar el modelo tanto con los datos de *train* como con los de *test*. Los parámetros γ_i , α_{0ij} y α_{1ij} de los sujetos de *test* deberían coincidir con los de personalización.

- Comprobación de que el MAE (*Mean Absolute Error*) está dentro de los valores esperados.

Estimación de GRACE

La estimación es el paso posterior a la personalización. Son necesarios los parámetros de la población, así como los individuales del sujeto, para estimar los valores de sus marcadores en un instante dado. Basta con aplicar la fórmula que modela las trayectorias de los marcadores:

$$Y_{ij} = g_j(t + \gamma_i) + \alpha_{0ij} + \alpha_{1ij}t$$

4.6.2. Leaspy

Antes de introducir los datos a Leaspy hay que asegurarse que los valores de la columna ID son de tipo string. Además, se fijarán las columnas ID y TIME como índices del set de datos. El modelo usado será *logistic* y con la aproximación `mcmc_saem` (*Markov chain Monte Carlo - Stochastic Approximation of Expectation Maximization*). El proceso consta de los siguientes pasos:

1. Crear el modelo de Leaspy con los parámetros definidos.
2. Ajustar el modelo a los datos mediante el método *fit*. Esto permitirá estimar la historia natural de la enfermedad.
3. Personalizar los resultados al set de datos mediante el algoritmo `scipy_minimize`. De esta forma se obtendrán las trayectorias individuales y los desplazamientos temporales de cada sujeto.
4. Estimar las trayectorias individuales en base a los parámetros obtenidos en la personalización.

Algoritmo 6 Preparación de los datos

Entrada:

- Tabla ADNIMERGE procesada
- Vector de características

Salida:

- Tabla con datos de entrenamiento
- 1: Leer tabla ADNIMERGE
 - 2: Seleccionar población preclínica (CU y SMC) y sMCI
 - 3: Filtrar columnas para quedarse con marcadores de interés
 - 4: Crear columnas para entrenar ambos algoritmos (p.ej. TIME, CONVERSION_TIME_DAYS ...)
 - 5: Convertir marcadores a escala percentil
 - 6: Eliminar sujetos con todos los valores NaN
 - 7: Eliminar sujetos con una única visita
 - 8: Guardar el modelo
-

4.6.3. Reducción del problema multidimensional a unidimensional

GRACE y *Leaspy* se entrenan con vectores de características de varias dimensiones, las cuales evolucionan con el tiempo. Con el objetivo de determinar el tiempo de conversión

de los sujetos y su diagnóstico, es necesario reducir el problema multidimensional a uno unidimensional. Esto facilitará posteriormente el entrenamiento del clasificador de Bayes y su aplicación. La evolución de la enfermedad y, por tanto, el desplazamiento temporal de los sujetos lo define el DPS (*Disease Progression Score*):

- GRACE: es el desplazamiento temporal del sujeto a largo plazo más el tiempo central a corto plazo:

$$DPS = \gamma_i + t_i^c$$

γ_i sigue una normal de la forma $\gamma_i \sim \mathcal{N}(0, \sigma_\gamma)$. Valores de γ_i mayores que 0 indican que el sujeto está más evolucionado que la media en la enfermedad, y viceversa.

- Leaspy: es la exponencial de ξ_i por la diferencia entre el desplazamiento temporal a corto plazo y el parámetro τ_i :

$$DPS = \exp(\xi_i)(t_i - \tau_i)$$

Los valores de ξ_i y τ_i para cada sujeto se obtienen en la personalización de Leaspy:

- ξ_i es el factor de aceleración: indica la velocidad de progresión del sujeto. Valores de ξ_i mayores que 1 indican que el sujeto evoluciona más rápido que la media. Valores menores de 1 indican lo contrario.
- τ_i es el desplazamiento temporal: retrasa la aparición de la enfermedad un determinado tiempo. Su distribución sigue una normal de la forma $\tau_i \sim \mathcal{N}(\bar{\tau}, \sigma_\tau^2)$, donde $\bar{\tau}$ es el desplazamiento temporal medio de la población y σ_τ la varianza. Si $\tau_i > \bar{\tau}$ la enfermedad empezará más tarde en el sujeto en cuestión, y viceversa.

4.7. Métodos no paramétricos de progresión de la enfermedad

Las redes neuronales recurrentes son modelos específicamente diseñados para trabajar con secuencias. Es decir, con datos que presentan un orden temporal. El procedimiento para su construcción es similar al de los métodos paramétricos, con alguna salvedad:

1. Preparación de los datos: se modifican los siguientes aspectos:

- La normalización de los datos se realiza mediante estandarización¹. Esto evita diferencias de escala que den lugar a características con mayor influencia.
- No es necesario que los marcadores evolucionen en el mismo sentido, ya que las redes neuronales no asumen la forma de las curvas².
- Alineación temporal: el intervalo entre visitas de las secuencias debe ser fijo. Si la separación entre estas es superior a dicho intervalo, se añaden visitas faltantes intermedias en los puntos correspondientes. Las observaciones que no sean múltiplo del paso elegido se descartan³. Como resultado se obtienen dos conjuntos de datos: el original y una réplica con los valores faltantes interpolados. Si no fuera posible interpolar algún dato, se imputaría con un valor por defecto. El criterio

¹Restar la media y dividir por la desviación típica.

²No se imponen curvas monotonamente crecientes, por lo que pueden detectar sujetos reversores.

³Por ejemplo, sea un sujeto con visitas en los meses 6, 9, 12 y 18. Si el paso es de 6 meses, los datos del mes 9 se eliminan.

para la imputación del diagnóstico se basa en considerarlo cognitivamente normal antes de la conversión y con deterioro cognitivo leve después⁴. En cualquier caso, el *baseline* siempre debe ser completo⁵.

- Ventana deslizante: las redes neuronales recurrentes se entrenan con secuencias de longitud fija, mientras que el número de visitas de cada sujeto es variable. Para solventar este problema, se establece un tamaño de secuencia determinado y una ventana deslizante de dicho tamaño recorre las observaciones los sujetos. La última entrada de cada secuencia corresponde a la salida del sistema. El resto son las entradas. Por ejemplo, sea un sujeto con los siguientes diagnósticos: CN, CN, CN, MCI, CN, MCI. Si la longitud de secuencia es 4 (en azul las entradas y en rojo las salidas), se tendrían 3 secuencias: [CN, CN, CN, MCI], [CN, CN, MCI, CN] y [CN, MCI, CN, MCI].

Los algoritmos 7 y 8 describen los procesos de alineación temporal y construcción de las secuencias.

2. Entrenamiento de los modelos: se modifican los siguientes aspectos:

- No se utiliza un clasificador de Bayes independiente para diagnosticar el estado clínico. La información del diagnóstico se incluye en las secuencias y se estima conjuntamente con los marcadores.
- No existen parámetros específicos de los sujetos que definen el proceso de personalización. Se entrena únicamente una red con los mismos pesos para cada sujeto. Las secuencias se introducen sucesivamente y se van obteniendo las estimaciones en los puntos temporales siguientes.

3. Comparación de resultados: no hay ningún cambio con respecto a los métodos paramétricos. Se utilizan las mismas métricas.

Durante el entrenamiento, las redes neuronales recurrentes buscan minimizar una función de pérdida que mide la similitud entre las salidas predichas y reales. Para tener en cuenta tanto variables categóricas como continuas, se pondera el MAE con la entropía cruzada binaria (BCE, *Binary Cross Entropy*). Únicamente se aplica sobre observaciones reales. Las imputaciones se enmascaran, de forma que quedan excluidas del cómputo.

El algoritmo 9 muestra el pseudo-código de la función de pérdida de ponderación del MAE y BCE.

4.8. Análisis masivo

El análisis masivo tiene por objetivo determinar el vector de características que mejor modela la enfermedad. Los modelos son entrenados para cada una de las posibles combinaciones de vectores, con dimensión máxima 5. El mejor modelo será aquel cuyos valores de detección de sujetos pCU y correlación con la reserva sean mayores. Los vectores se forman combinando los siguientes marcadores:

- RAVLT Forgetting

⁴Los sujetos sCU no han convertido todavía y los sMCI convirtieron antes de la primera visita, por lo que se imputan con diagnóstico CN y MCI, respectivamente.

⁵En caso de no serlo, se imputará con el valor por defecto.

⁶Las máscaras indican las posiciones de las observaciones con un 1 y de los valores faltantes con un 0. El sumatorio de una máscara coincide con el total de observaciones. En el cálculo del MAE y BCE se añade una constante de reducido valor en el denominador para evitar divisiones por 0 en el caso de no haber valores faltantes.

Algoritmo 7 Alineación temporal

Entrada:

- Tabla ADNIMERGE
- Longitud de la secuencia
- Paso temporal
- Valor de imputación por defecto

Salida:

- Tabla con las observaciones alineadas temporalmente
- Tabla anterior con los valores faltantes interpolados (o imputados)

```

1: Eliminar los sujetos con menos visitas que la longitud de secuencia
2: for sujeto en sujetos do
3:   Determinar el mes de la última visita (mes_ultimo)
4:   Obtener los meses entre el baseline y mes_ultimo, múltiplos del paso
5:   Eliminar las visitas que no estén en los meses calculados
6:   Introducir visitas faltantes en los meses no observados
7:   if existen valores faltantes then
8:     if diagnóstico then
9:       if la visita ocurre antes de la conversión then
10:        Imputar con CN
11:      else
12:        Imputar con MCI
13:      end if
14:    else
15:      Interpolar los valores faltantes
16:      if excepción then
17:        Imputar los valores faltantes con el valor por defecto
18:      end if
19:    end if
20:  end if
21: end for

```

Algoritmo 8 Construcción de las secuencias

Entrada:

- Tabla con las observaciones alineadas temporalmente
- Tabla anterior con los valores faltantes interpolados (o imputados)
- Longitud de la secuencia (`longitud_secuencia`)

Salida:

- Secuencias de entradas con y sin interpolación
 - Salidas de las secuencias con y sin interpolación
 - Posiciones de los valores faltantes en las secuencias de entrada
 - Posiciones de los valores faltantes en las secuencias de salida
- 1: Dividir los datos en *train*, *test* y validación
 - 2: Estandarizar los conjuntos con las medias y desviaciones típicas de *train*
 - 3: **for** sujetos en [sujetos originales; sujetos interpolados] **do**
 - 4: **for** sujeto en sujetos **do**
 - 5: Calcular el número de visitas (`nvisitas`)
 - 6: **for** i en 0 : `nvisitas` **do**
 - 7: **if** i + `longitud_secuencia` \neq `nvisitas` **then**
 - 8: `secuencia` = `sujeto`[i : i + `longitud_secuencia`]
 - 9: Añadir `secuencia` a la lista de secuencias
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: **end for**
 - 14: Obtener las salidas como la última fila de cada secuencia
 - 15: Obtener las secuencias de entrada (las filas que no son salidas)
 - 16: Determinar las posiciones de los valores faltantes en las entradas y salidas sin interpolar
-

Algoritmo 9 Función de pérdida con ponderación del MAE y BCE

Entrada:

- Salida de la LSTM (\hat{y})
- Salida real esperada (y)
- Máscara de valores faltantes (m)
- Pesos del MAE y BCE (α_{mae} y α_{bce})

Salida:

- Valor de pérdida
- 1: Determinar las salidas continuas: $\hat{y}_{mae} = \hat{y}[: -1]$, $y_{mae} = y[: -1]$
 - 2: Determinar las salidas categóricas: $\hat{y}_{bce} = \hat{y}[-1]$, $y_{bce} = y[-1]$
 - 3: Determinar las máscaras continua y categórica⁶: $m_{mae} = m[: -1]$, $m_{bce} = m[-1]$
 - 4: Calcular: $MAE = \frac{\sum m_{mae} \odot |\hat{y}_{mae} - y_{mae}|}{\sum m_{mae}}$
 - 5: Calcular: $BCE = \frac{\sum m_{bce} \odot [-y_{bce} \log(\hat{y}_{bce}) - (1 - y_{bce}) \log(1 - \hat{y}_{bce})]}{\sum m_{bce}}$
 - 6: Calcular la pérdida ponderando el MAE y BCE: $\mathcal{L} = \alpha_{mae} \cdot MAE + \alpha_{bce} \cdot BCE$
-

- RAVLT Immediate
- RAVLT Learning
- RAVLT Perc Forgetting
- ADAS13
- FAQ
- MMSE
- CDRSB
- LDELTOTAL
- mPACCdigit
- mPACCtrailsB
- TRABSCOR

Únicamente se han utilizado *tests* neuropsicológicos, ya que son más baratos, sencillos y rápidos de implementar en la práctica clínica frente a otras pruebas (por ejemplo, resonancias magnéticas). Además, estas últimas no presentan mejoras significativas en los resultados.

A continuación, se describe el procedimiento utilizado (archivos `multigrace.R`, `multileaspy.py` y `multirnn.py` de `/demo/multigrace/`, `/demo/multileaspy/` y `/demo/multirnn/`, respectivamente):

1. Lectura de los datos de ADNI, filtrado y normalización por escala percentil o estandarización, según corresponda. División en *train* y *test*. División de los datos en entrenamiento y validación⁷, alineación temporal y construcción de las secuencias en el caso de las redes neuronales.
2. Generación de las tablas con los datos necesarios para la construcción de los modelos. Cada entrada incluye la combinación de marcadores con su identificador, el sentido de las puntuaciones, y el directorio donde se alojan los datos de entrada y salida. Las entradas son ficheros `.xlsx` con los datos necesarios para el entrenamiento de cada combinación. Las salidas son tantos modelos por combinación como se especifique en el *bootstrap*. También se guarda un fichero donde se registran los resultados de cada modelo, así como los identificadores de los sujetos con los que han sido entrenados.
3. Entrenamiento y obtención de las medidas de calidad de los modelos. Son responsables los ficheros `build_grace.R`, `build_leaspy.py` y `build_lstm.py`, situados en `/demo/multigrace/aux`, `/demo/multileaspy/aux` y `/demo/multirnn/aux`, respectivamente. Consta de 3 pasos:
 - a) Construcción de los modelos: se realiza con las llamadas a `buildModelGrace`, `buildModelLeaspy` y `buildModelLSTM` de los ficheros `build_model_grace`, `build_model_leaspy`, y `build_model_lstm` situados en la misma carpeta. En el caso de los métodos paramétricos, estas funciones realizan el proceso de *bootstrap*, que proporciona n combinaciones de sujetos para construir n modelos a partir de un mismo *dataset* de entrenamiento. También personalizan los datos de *train*

⁷Tantas veces como se indique en el parámetro de *bootstrap*.

y obtienen el DPS.

Los modelos GRACE se guardan en un fichero `.rds`, mientras que se utiliza el formato `.pk1` para Leaspy y LSTM.

- b) Construcción del clasificador de Bayes (solamente métodos paramétricos): con el DPS de los sujetos de entrenamiento se construyen n clasificadores de Bayes, uno por cada *bootstrap* realizado. La estimación de la función de densidad del DPS se hace con la técnica de KDE. El ancho de banda se calcula con la regla empírica de Scott. Los sujetos se clasifican en CN o MCI.
- c) Obtención de las medidas de calidad: se encargan las funciones `getAUC`, `getMAE` y `getOnset`, situadas en `/demo/multigrace/aux`, `/demo/multileaspy/aux` y `/demo/multirnn/aux`.

En los métodos paramétricos, el AUC se calcula a partir del DPS, el clasificador de Bayes y los diagnósticos reales. El MAE requiere estimar los valores de los marcadores en los momentos que se realizan las visitas y antitransformarlos de vuelta a escala original. En `getOnset` se calculan las *accuracies* de sCU, pCU y sMCI, y las correlaciones con la edad y la reserva.

En las redes neuronales, es preciso desestandarizar las salidas. Los diagnósticos se estiman directamente.

Los resultados obtenidos para cada *bootstrap* se guardan en un fichero y se devuelven como argumento de `return` de `buildGrace`, `buildLeaspy` y `buildLSTM`.

4. Almacenamiento de resultados: se reciben las medidas de calidad para cada combinación de marcadores y se calcula la media y desviación típica de los n *bootstrap*. Se componen 2 tablas con los resultados de todas las combinaciones realizadas: una para el AUC, las correlaciones y las *accuracies*, y otra para el MAE. Las tablas se guardan en ficheros `.xlsx`: la primera en `preclinical_CAM_jul22_1_2_3_4_5_6_7_8_9_10_11_12.xlsx`, y la segunda en `MAE_test_preclinical_CAM_jul22.xlsx`.

El pseudo-código de este procedimiento se muestra en el algoritmo 10.

Los algoritmos se ejecutan en Magerit-3, habiéndose paralelizado previamente para explorar la capacidad de cómputo del *cluster*. La paralelización se realiza usando el estándar MPI (*Message Passing Interface*), mediante una arquitectura maestro-esclavo. Se usa el paquete `Rmpi` para `multiGrace`, y `mpi4py` para `multiLeaspy` y `multiLSTM`. El código para la paralelización se encuentra en `multigrace_par_mpi.R`, `multileaspy_par_mpi.py` y `multirnn_par_mpi.py` de `/demo/multigrace/`, `/demo/multileaspy/` y `/demo/multirnn/`, respectivamente.

4.8.1. Medidas de calidad

La capacidad que tiene un modelo de representar el desarrollo de la enfermedad se evalúa con las medidas de calidad. Se consideran las siguientes:

- Porcentaje de sCU: la cantidad de sujetos clasificados como sCU correctamente frente al número de sujetos sCU reales, expresado en tanto por ciento.
- Porcentaje de pCU: la cantidad de sujetos clasificados como pCU correctamente frente al número de sujetos pCU reales, expresado en tanto por ciento.

Algoritmo 10 multiGrace, multiLeaspy y multiLeaspy

Entrada:

- Tabla ADNI

Salida:

- Fichero preclinical_CAM_jul22_1_2_3_4_5_6_7_8_9_10_11_12.xlsx
 - Fichero MAE_test_preclinical_CAM_jul22.xlsx
- 1: Leer tabla ADNI, filtrar datos, normalizar los datos (escala percentil o estandarización)
 - 2: **if** el modelo es una red neuroanal recurrente **then**
 - 3: Alineación temporal y construcción de las secuencias
 - 4: **end if**
 - 5: Generar tablas de combinaciones de marcadores y argumentos para buildGrace, buildLeaspy y buildLSTM
 - 6: **for** combinación en combinaciones **do**
 - 7: **for** modelo en *bootstrap* **do**
 - 8: Construir modelo con los datos de entrenamiento
 - 9: Guardar el modelo
 - 10: **if** el modelo es paramétrico **then**
 - 11: Personalizar a los sujetos
 - 12: Obtener el DPS
 - 13: Construir el clasificador de Bayes
 - 14: **end if**
 - 15: Obtener AUC, MAE, *accuracies* y correlaciones
 - 16: **end for**
 - 17: Media y desviación típica de las medidas de calidad para los n *bootstrap*
 - 18: **end for**
 - 19: Almacenamiento de los resultados en preclinical_CAM_jul22_1_2_3_4_5_6_7_8_9_10_11_12.xlsx y MAE_test_preclinical_CAM_jul22.xlsx
-

- Porcentaje de sMCI: la cantidad de sujetos clasificados como sMCI correctamente frente al número de sujetos sMCI reales, expresado en tanto por ciento.
- Área bajo la curva (AUC): permite comparar la capacidad de clasificación de los modelos. Cuanto más cercano a 1 sea su valor, mayor capacidad de predicción tiene el modelo. Un clasificador aleatorio es aquel cuyo $AUC = 0.5$.
- Correlación con la edad: regresión lineal entre la edad de conversión real de los sujetos y la edad estimada.
- Correlación con la reserva: regresión lineal entre el tiempo real y el estimado para que los sujetos conviertan a MCI. Su valor es más bajo que el de la correlación con la edad, debido a que el rango de valores es más reducido.
- *Mean Absolute Error*: error medio absoluto entre el valor de los marcadores en una determinada visita según el modelo y el valor real.

El diagnóstico en una determinada visita se determina con un clasificador de Bayes. Para ello, es necesario estimar el DPS en el momento de la visita. La edad de conversión y la reserva se calcula con los sujetos pCU. El sujeto convierte en el día intermedio entre la primera visita que es diagnosticado como MCI y la visita anterior, diagnosticado como CN.

4.8.2. Partición de los datos

La evaluación de los modelos se realiza dividiendo el conjunto de datos total en dos subconjuntos que no se superponen: *test* y *train*. Este último se divide, a su vez, en una partición de entrenamiento y otra de validación:

- Métodos paramétricos: se emplea *bootstrapping* en cada combinación de características. En cada iteración del *bootstrapping* se selecciona aleatoriamente y con reemplazo un subconjunto de datos de entrenamiento. Al realizar la selección, se tiene en cuenta el diagnóstico clínico para asegurar que los datos están balanceados. Con esto se logra una evaluación más robusta de los modelos, al considerar la variabilidad que se introduce al utilizar distintos conjuntos de datos de entrenamiento.
- Redes neuronales recurrentes: las mismas particiones se utilizan en todas las combinaciones de características. Los sujetos se seleccionan de forma aleatoria y sin reemplazo. Dado que las redes se alimentan de secuencias de longitud fija, el número de visitas y el grupo clínico no son parámetros relevantes en el particionado⁸.

4.9. Robustez frente a valores faltantes

La necesidad de evaluar la robustez de los modelos surge de la falta de algunos marcadores durante las visitas de los pacientes. El modelo se evalúa para distintos porcentajes de valores faltantes. Esto se logra reemplazando aleatoria y equitativamente ciertos valores por NaN en cada marcador, hasta alcanzar el porcentaje definido. Al menos debe conservarse un valor de los marcadores en cada visita, a fin de no invalidarla.

El algoritmo 11 describe el proceso.

⁸Por ejemplo, sea un sujeto conversor con 5 visitas: CN, CN, CN, CN, MCI. Si la longitud de secuencia es 4 (entradas en azul y salida en rojo) se tendrían 2 secuencias: [CN, CN, CN, CN] y [CN, CN, CN, MCI].

El código se encuentra en las carpetas `/demo/multileaspy` y `/demo/multigrace`, ejecutando los ficheros `multileaspy.py` y `multigrace.R`, respectivamente; variando el porcentaje de valores faltantes a introducir.

Algoritmo 11 Reemplazamiento por NaN

Entrada:

- Tabla ADNIMERGE procesada
- Vector de características
- Porcentaje de NaN deseado

Salida:

- Tabla ADNIMERGE con un porcentaje de NaN
- 1: Calcular el número total de visitas
 - 2: Calcular el porcentaje inicial de NaN
 - 3: Calcular el número de visitas a reemplazar por marcador (`v_x_marcador`)
 - 4: **for** marcador en marcadores **do**
 - 5: `no_nan` = visitas distintas de NaN
 - 6: `nan_sig` = visitas con NaN en el siguiente marcador
 - 7: `nan_ant` = visitas con NaN en el anterior marcador
 - 8: `no_nan` = (`no_nan` \ `nan_sig`) \ `nan_ant`
 - 9: Reemplazar `v_x_marcador` valores en el conjunto `no_nan` por NaN
 - 10: **end for**
-

4.10. Análisis frente al número de visitas

Se analiza el desempeño de los DPM agrupando los sujetos según su número de visitas. Las medidas de calidad empleadas son las mismas que en el análisis masivo⁹. Cuantos más sujetos conversores con pocas visitas o estables con muchas visitas sea capaz de detectar el modelo, más fiable será. La personalización se realiza con todas las visitas disponibles de cada paciente. La población utilizada es el conjunto combinado de *train* y *test*.

El algoritmo 12 resume el procedimiento.

El código se encuentra en la carpeta `/demo/analisis_numero_visitas/`, ejecutando los ficheros `n_visits_analysis_leaspy.py` y `n_visits_analysis_grace.R`. El fichero `n_visits_compare.py`, situado en la misma carpeta, grafica los resultados obtenidos.

4.11. Pronóstico

Hasta ahora, los DPM se han empleado en tareas de diagnóstico. Es decir, la clasificación y estimación de los tiempos de conversión se realiza en un intervalo temporal en el que los resultados de todas las visitas son conocidos. En esta sección, se pretende evaluar la capacidad de predicción de los modelos. Esto implica que las últimas visitas de los sujetos dejarán de ser conocidas, pasando a ser estimadas y evaluadas por los DPM.

El código se encuentra en la carpeta `/demo/pronostico/`, ejecutando los ficheros

⁹AUC, %sCU, %pCU, %sMCI, correlación con la edad y con la reserva, MAE.

Algoritmo 12 Análisis frente al número de visitas con GRACE / Leaspy

Entrada:

- Mejores DPM con GRACE / Leaspy
- Clasificador de Bayes
- Vector con número de visitas a analizar (vector_visitas)

Salida:

- Tabla con medidas de calidad frente al número de visitas
- 1: Cargar los mejores DPM de GRACE / Leaspy
 - 2: **for** n_visitas en vector_visitas **do**
 - 3: Buscar los sujetos con n_visitas (sujetos_n)
 - 4: **for** sujeto en sujetos_n **do**
 - 5: Personalización del sujeto
 - 6: Obtener las medidas de calidad
 - 7: **end for**
 - 8: **end for**
 - 9: Devolver tabla con medidas de calidad frente al número de visitas
-

future_leaspy_prediction.py y future_grace_prediction.R. El fichero future_prediction_compare.py, situado en la misma carpeta, grafica los resultados obtenidos.

4.11.1. Modelos de progresión de la enfermedad

Los modelos a utilizar son los considerados como mejores provenientes del análisis masivo. Se sitúan en la carpeta /demo/pronostico/data y consituyen la entrada del algoritmo. Cada modelo consiste en 10 modelos entrenados con distintos subconjuntos de datos de entrenamiento según el procedimiento de *bootstrapping*.

El procedimiento consiste en eliminar sucesivamente las últimas visitas de los sujetos, realizando la personalización con el resto. Se impone como condición de diseño que la mitad del total de las visitas de cada sujeto debe ser mayor o igual al número de visitas a predecir. Así, para un sujeto con 5 visitas solamente podrán estimarse la 4 y la 5, empleando las visitas 1 a 3 y 1 a 4, respectivamente, para la personalización. En cada visita se estiman los valores futuros de los marcadores y se calcula el error absoluto. Adicionalmente, se emplea el clasificador de Bayes obtenido en el análisis masivo para determinar el diagnóstico futuro en función del DPS estimado.

Para cada visita estimada, se calcula el tiempo transcurrido desde la última visita de personalización¹⁰. Esto permite agrupar las estimaciones por cercanía temporal. Cada grupo (1 año en el futuro, 2 años en el futuro...) debe contar con al menos 30 muestras para ser considerado.

Especial importancia cobra la capacidad de los modelos de detectar sujetos pCU con antelación y obtener su tiempo de conversión. Sobre este grupo clínico, se eliminan todas las visitas posteriores a la primera en la cual son diagnosticados MCI. A continuación, se van eliminando las últimas visitas restantes siguiendo el procedimiento anterior (i.e. imponiendo la misma condición de diseño¹¹). La personalización se lleva a cabo con las

¹⁰Nótese que el tiempo entre visitas no es regular. Por ejemplo, si se elimina una visita, el tiempo entre esta y la última de personalización puede ser variable entre sujetos.

¹¹La mitad del total de visitas antes del primer diagnóstico MCI, contando esta, debe ser mayor o igual al número de

visitas que no han sido eliminadas. Posteriormente, se realiza una estimación entre la primera visita del sujeto y 10 años después de la primera en la que fue diagnosticado MCI, con una resolución de 0.2 años. El instante en el que la probabilidad de MCI estimada por el clasificador de Bayes sea mayor a 0.5, coincidirá con el instante de la conversión¹². De esta forma, se determina el tiempo de conversión con exactitud.

El algoritmo 13 resume ambos procedimientos.

Algoritmo 13 Pronóstico con GRACE / Leaspy

Entrada:

- Mejores DPM con GRACE / Leaspy
- Clasificador de Bayes
- Vector con número de visitas a predecir (`visitas_futuro`)

Salida:

- Tabla con MAE y diagnóstico futuro
 - Tabla con tiempos de conversión de sujetos pCU
- 1: Cargar los mejores DPM de GRACE / Leaspy
 - 2: **for** `n_visitas_futuro` en `visitas_futuro` **do**
 - 3: **for** sujeto en sujetos **do**
 - 4: **if** `n_visitas_sujeto / 2 >= n_visitas_futuro` **then**
 - 5: Estimar visitas del sujeto
 - 6: **end if**
 - 7: **if** sujeto es pCU **then**
 - 8: Eliminar todas las visitas antes de la primera con diagnóstico MCI
 - 9: **if** `n_visitas_estables_sujeto / 2 >= n_visitas_futuro` **then**
 - 10: Estimar trayectoria entre la primera visita y 10 años después de la primera con diagnóstico MCI con resolución 0.2 años
 - 11: Determinar tiempo de conversión
 - 12: **end if**
 - 13: **end if**
 - 14: **end for**
 - 15: **end for**
 - 16: Devolver tabla con MAE y diagnóstico futuro
 - 17: Devolver tabla con tiempos de conversión de sujetos pCU
-

4.11.2. Predicción constante

Además de los algoritmos propuestos hasta el momento, se incluye la predicción constante. Se trata de un algoritmo que no requiere entrenamiento¹³, y su estimación futura consiste en el valor de la última visita observada. La predicción constante, por tanto, debe cumplir:

- La detección de sujetos sCU debe ser del 100 %¹⁴: la propia definición de sCU (aque-

visitas eliminadas. Por ejemplo, si un sujeto presenta los siguientes diagnósticos: CU, CU, CU, CU, MCI, CU, MCI, MCI, solamente se tienen en cuenta las 5 primeras visitas. En este caso, la estimación solo se podría realizar con las visitas 1 a 3 y 1 a 4.

¹²Se recuerda que el tiempo de conversión se calcula como la media entre la edad de conversión a MCI y la edad en la visita anterioro.

¹³Nótese que no es necesario *bootstrapping*.

¹⁴Lo mismo debería ocurrir con la detección sMCI. Sin embargo, al no haberse incluido en el estudio los grupos rMCI y pMCI, hay un pequeño descenso en esta métrica.

Los sujetos que su la última visita de personalización coincide con la visita de predicción) describe el funcionamiento del algoritmo.

- La detección de sujetos pCU debe ser del 0%: al estimar el valor de la última visita de personalización no es capaz de detectar cambios en el diagnóstico.

4.11.3. Partición de los datos

Las predicciones se realizan con el total de los datos de *train* y *test*. De esta forma, el número de muestras en cada año futuro estimado aumenta. Aunque pueda aumentar el riesgo de *overfitting* al usar las visitas de *train*, cabe recalcar que cada modelo solamente ha visto una fracción de estas visitas debido al *bootstrap*.

Por otro lado, es posible aumentar la cantidad de muestras prediciendo todas las visitas futuras en lugar de únicamente la última correspondiente a cada iteración. Por ejemplo, en un sujeto con las siguientes visitas (en rojo, verde, azul la primera, segunda, tercera visitas a predecir, respectivamente): CU, CU, MCI, CU, CU, MCI, CU. Al predecir la segunda visita, se predice también la primera. Al predecir la tercera visita, se predicen también la primera y la segunda. En este caso, se ha triplicado el número de muestras. Es necesario enfatizar que, a pesar de que la primera visita se esté prediciendo 3 veces, cada vez se utiliza un número distinto de visitas en la personalización. A tal efecto, se consideran como muestras distintas.

4.11.4. Medidas de calidad

Las medidas de calidad permiten evaluar las estimaciones de pronóstico de los modelos. A diferencia del diagnóstico en donde únicamente se tenían en cuenta la primera y última visita observadas, en pronóstico se debe tener en cuenta también el diagnóstico en la visita que se está prediciendo. Esto fuerza a una redefinición de los grupos clínicos:

- sCU: tanto en la primera como la última visita de la personalización el sujeto es CU. En la visita de la predicción, el sujeto también es CU. Es decir, se trata de sujetos que son CU durante la personalización y lo siguen siendo en la predicción. Si en la última visita de la personalización fuera MCI, el sujeto ya habría convertido durante la personalización, y posteriormente habría revertido. En tal caso, debería clasificarse dentro de un nuevo grupo que no se ha tenido en cuenta en el estudio: rMCI (sujetos MCI que revierten a CU). Por este motivo, se impone la condición de que la última visita de la personalización tenga que ser CU.
- pCU: tanto en la primera como la última visita de la personalización el sujeto es CU. En la visita de la predicción, el sujeto es MCI. Siguiendo el razonamiento del caso anterior, son sujetos estables durante la personalización, pero que convierten en el futuro. Uno podría pensar que la última visita de la personalización debería ser MCI. Si así fuera, el sujeto ya habría convertido durante la personalización, y seguiría siendo MCI en el futuro. Por lo tanto, sería ilógico estimar su conversión en el futuro, ya que se trataría de un sujeto estable MCI.
- sMCI: se diferencian 2 casos:
 - El sujeto es MCI en la primera visita de la personalización y en la visita de la predicción. Si en la última visita de la personalización el diagnóstico fuera CU, pertenecería al grupo de los rMCI. Si fuera Demencia, sería pMCI (sujetos MCI

progresivos a demencia). Debido a que estos grupos no se han considerado en el estudio, en estos casos no se tiene en cuenta la última visita de la personalización.

- El sujeto es CU en la primera visita de la personalización, y MCI en la última visita de la personalización y la visita de la predicción. Aunque de primeras pudiera pensarse que se trata de un sujeto pCU, debe entenderse que la conversión ya ha ocurrido y a partir de ese momento el sujeto es MCI. Visto de otra forma, si fuera clasificado como pCU, se estaría intentando predecir la conversión de un sujeto que ya se sabe que ha convertido, resultando en una contradicción.

Cabe mencionar que a según se van realizando las predicciones en distintas visitas, el grupo clínico de los sujetos puede variar. Suponiendo un sujeto con los siguientes diagnósticos (siendo los de color rojo los que se quiere estimar): CU, CU, MCI, CU, CU, MCI, CU. Este sujeto se clasificaría como sCU, pCU, sCU, respectivamente.

Con estas nuevas definiciones, se evalúan los modelos según las siguientes medidas de calidad: %sCU, %pCU, %sMCI, AUC, MAE. Adicionalmente, se introduce una nueva medida: BCA (*Balanced Accuracy*).

Las correlaciones con la edad y la reserva se calculan según se ha expuesto anteriormente: dentro de los sujetos pCU reales, se calcula su tiempo de conversión considerando para la personalización únicamente las visitas anteriores a la primera visita con diagnóstico MCI. A tales efectos, se extiende el período de estimación 10 años después de la primera visita MCI.

4.12. Historia natural

Con los DPM seleccionados en el análisis masivo se reconstruyen las trayectorias de los sujetos, permitiendo obtener el tiempo de conversión. De esta forma, es posible situar a los sujetos en la escala temporal natural¹⁵ y conocer la progresión de los marcadores más allá de los límites de las visitas.

Los sujetos pCU convierten durante las visitas, por lo que se puede saber su tiempo de conversión real. Sin embargo, los sujetos sCU y sMCI no pueden situarse directamente sobre la escala temporal. Los primeros no han convertido aun después de la última visita. Los segundos ya habían convertido antes de la primera visita. El método consiste en estimar las trayectorias 10 años antes y 10 años después de la primera y la última visita de cada paciente, respectivamente. La resolución es de 0.2 años. Esto permite determinar cuándo convertirán los sCU y cuándo lo hicieron los sMCI. Los sujetos que no conviertan en el intervalo de tiempo extendido no se consideran en el análisis.

A partir del ordenamiento temporal, es posible ajustar la curva que minimiza el error cuadrático de los marcadores, obteniendo la historia natural del marcador. Este ajuste no se limita únicamente a los vectores de características de entrenamiento, sino que también permite representar otros marcadores¹⁶.

El proceso se detalla en el algoritmo 14.

¹⁵Hasta ahora se habían representado en la escala DPS.

¹⁶Incluidos de tipo CSF, T1-MRI o PET. No tienen por qué ser de tipo NM.

El código se encuentra en la carpeta `/demo/historia_natural_dpm/`, ejecutando los ficheros `estimateConversionTime.py` y `estimateConversionTime.R`. Los ficheros `plotReserveDistribution.py` y `plotTimeline.py`, situados en la misma carpeta, grafican histogramas con los tiempos de conversión y las historias naturales de diversos marcadores, respectivamente.

4.12.1. Leaspy y GRACE

La curva que se pretende ajustar es la logística propuesta en [38]:

$$\gamma_k(t) = (a - d) \left(1 + \left(\frac{1}{p_k} - 1 \right) \exp \left(-\frac{v_k}{p_k(1 - p_k)}(t - t_0) \right) \right)^{-1} + d$$

Mediante un proceso de optimización¹⁷ se obtienen los valores de los parámetros p_k , v_k y t_0 . Se establecen los límites inferiores $[0, 0, -20]$ y superiores $[1, \infty, 20]$, respectivamente. A continuación, se explica el significado de estos parámetros:

- p_k : valor de la ordenada en el tiempo t_0 . Queda restringido entre 0 y 1 al tratarse de una logística.
- v_k : derivada de la curva en el punto t_0 . Se limita entre 0 e ∞ , ya que la velocidad puede alcanzar cualquier valor y las curvas son monotonamente crecientes.
- t_0 : desplazamiento temporal de la curva. Se limita entre -20 y 20 , aunque podría ampliarse este intervalo.

Los valores iniciales de los parámetros son $[0,5, 0,5, 0,5]$. Los valores de a y d representan los límites reales de las pruebas, es decir, los valores entre los cuales están comprendidas las logísticas. Por ejemplo, para ADAS13 se establece $a = 85$ y $d = 0$.

4.12.2. RPDPM

La curva se rige por las ecuaciones propuestas en [31]:

$$f(s; \theta) = (a - d) \left(1 + \frac{1}{\gamma} e^{-\frac{b}{\gamma}(s-c)} \right)^{-\gamma} + d$$

Donde b , c y γ definen los parámetros a ajustar. Se establecen los límites inferiores y superiores a $[0, -\infty, 0]$ y $[\infty, \infty, \infty]$, respectivamente:

- b : tasa de crecimiento. Modula la velocidad con la que se pasa del límite superior al inferior. Se restringe a valores positivos por motivos de estabilidad.
- c : punto de inflexión de la curva. Puede tomar cualquier valor del conjunto de los números reales.
- γ : parámetro de simetría. Se tienen varias situaciones:
 - $\gamma = 1$: la curva es simétrica respecto al punto medio.
 - $\gamma \rightarrow 0^+$: el crecimiento se acelera después de alcanzar el punto medio.
 - $\gamma > 1$: se alcanza rápidamente el punto medio, mientras que la curva crece lentamente hasta el límite superior.

No se pueden tomar valores negativos de γ , ya que la función dejaría de ser acotada.

Los valores iniciales de los parámetros son $[1, 1, 1]$.

¹⁷Se usa la función `least_squares` de `scipy.optimize` de Python.

Algoritmo 14 Historia natural con GRACE / Leaspy

Entrada:

- Mejores DPM con GRACE / Leaspy
- Clasificador de Bayes
- Años de extensión del intervalo de las visitas y resolución

Salida:

- Tabla con tiempos de conversión de todos los sujetos
 - Trayectoria a largo plazo con la historia natural de los marcadores
 - Trayectoria a corto plazo con las visitas de los sujetos
- 1: Cargar los mejores DPM de GRACE / Leaspy
 - 2: Crear los intervalo de estimación entre [Primera visita - n años; Última visita + m años; Resolución]
 - 3: Estimar el DPS y la probabilidad de ser MCI en cada punto del intervalo
 - 4: Determinar la edad de conversión y la reserva
 - 5: **for** característica en características **do**
 - 6: Ordenar las observaciones de todos los sujetos según la reserva
 - 7: Ajustar los parámetros de la curva a largo plazo que menor error cuadrático cometen
 - 8: **for** sujeto en sujetos **do**
 - 9: Ordenar las observaciones del sujeto según la reserva
 - 10: Ajustar los parámetros de la curva a corto plazo que menor error cuadrático cometen
 - 11: **end for**
 - 12: **end for**
 - 13: Devolver tabla con tiempos de conversión de todos los sujetos
 - 14: Representar trayectorias a corto y largo plazo
-

4.13. Arquitectura maestro-esclavo: paralelización con MPI

La paralelización de `multiGrace`, `multiLeaspy` y `multiLSTM` en Linux se realiza usando el estándar MPI. Al ejecutar `mpirun` (Curie) o `srun` (Magerit-3), varios procesos ejecutan el mismo *script* dentro del mismo comunicador. Los procesos deben estar coordinados entre sí para garantizar una distribución eficiente de la carga de trabajo, evitando procesos ociosos o problemas de redundancia. La arquitectura maestro-esclavo responde adecuadamente a esta situación. Se distinguen 2 tipos de proceso:

- Maestro: es conocedor de las tareas a realizar y las distribuye dinámicamente entre los esclavos. En los casos de `multiGrace`, `multiLeaspy` y `multiLSTM`, genera las tablas con todas las posibles combinaciones de marcadores y los argumentos de las funciones que realizan los esclavos.
- Esclavo: ejecuta las tareas encomendadas por el maestro. En los casos de `multiGrace`, `multiLeaspy` y `multiLSTM`, cada tarea consiste en entrenar un modelo y obtener sus medidas de calidad.

El maestro es el proceso cuyo identificador dentro del comunicador predeterminado es 0. El resto de procesos son esclavos. Inicialmente, los esclavos están inactivos, esperando a que el maestro genere las tareas a realizar. Seguidamente, el maestro envía las tareas a los esclavos, activando una *flag* cada vez que la tarea sea recibida. Esta *flag* indica que el esclavo en cuestión está ocupado y no se le pueden mandar más tareas. Hay tantas *flags* como esclavos se hayan creado. Cuando un esclavo termina su tarea, el maestro desactiva la *flag* que lo identifica, quedando libre para realizar la siguiente tarea que le mande el

maestro. Si el maestro envía un código de finalización (por ejemplo, -1), el esclavo queda libre y avisa al maestro previamente a su finalización. De esta forma se asegura que el maestro sea el último proceso en terminar.

El algoritmo 15 muestra el pseudo-código del esquema descrito.

4.13.1. Problemas de memoria en Tensorflow

Por cada tarea que manda el maestro, los esclavos realizan un proceso de validación cruzada, lo que implica la construcción de varios modelos. Tras varias tareas, es posible encontrarse con un error parecido al siguiente:

slurmstepd: error: Detected 1 oom-kill event(s) in StepId=778254.0 cgroup. Some of your processes may have been killed by the cgroup out-of-memory handler.

Este mensaje indica que el sistema detectó un exceso de consumo de memoria RAM, y terminó uno o varios procesos como medida de protección. El problema se produce cuando se entrenan modelos en Tensorflow en bucle sin liberar correctamente la memoria. Consecuentemente, la memoria se acumula progresivamente hasta alcanzar el límite permitido por SLURM.

Una posible solución es utilizar `K.clear_session()` y borrar las referencias al modelo después de cada iteración de la validación cruzada. Sin embargo, no siempre funciona correctamente [6] [9]. Una solución alternativa consiste en encapsular la validación cruzada en un proceso independiente usando `subprocess`. De esta forma, el sistema operativo limpia los recursos no liberados dentro del proceso, y los esclavos ejecutan cada tarea sin residuos de tareas anteriores.

El algoritmo 16 muestra el pseudo-código modificado para evitar errores de memoria.

Algoritmo 15 Arquitectura maestro-esclavo para la paralelización de tareas usando MPI

```

1: pid = Identificador del proceso dentro del comunicador
2: if pid == 0 then
3:   Generación de las tareas a realizar
4:   for tarea en tareas do
5:     if todos los esclavos ocupados then
6:       Esperar la finalización de algún esclavo
7:       Actualizar el estado del esclavo a inactivo
8:       Agregar el resultado del esclavo a los resultados globales
9:     end if
10:    Mandar la siguiente tarea al primer esclavo libre
11:    Actualizar el estado del esclavo a ocupado
12:  end for
13:  for esclavo en esclavos do
14:    Mandar código de finalización al esclavo
15:    Recibir confirmación de finalización
16:  end for
17:  Guardar los resultados globales
18: else
19:   while 1 do
20:     Esperar tarea del maestro
21:     if tarea == código de finalización then
22:       break
23:     else
24:       Ejecución de la tarea
25:       Mandar los resultados al maestro
26:     end if
27:   end while
28:   Avisar al maestro de la finalización
29: end if

```

Algoritmo 16 Revisión del algoritmo 15 para evitar acumulación de memoria por modelos consecutivos entrenados con Tensorflow

```

1: pid = Identificador del proceso dentro del comunicador
2: if pid == 0 then
3:   ...
4: else
5:   while 1 do
6:     Esperar tarea del maestro
7:     if tarea == código de finalización then
8:       break
9:     else
10:      if carpeta de la tarea vacía then
11:        Guardar los argumentos de la tarea en la carpeta correspondiente
12:        Lanzar un proceso con el script a ejecutar y la ruta a los argumentos como entradas
13:        INICIO DEL SUBPROCESO
14:         1: Cargar los argumentos guardados de la ruta recibida por línea de comandos
15:         2: Ejecución de la tarea (validación cruzada de una combinación)
16:         3: Guardar los resultados del modelo
17:        FIN DEL SUBPROCESO
18:      end if
19:      Esperar la ejecución de la tarea por el nuevo proceso
20:    end if
21:    Cargar los resultados del modelo
22:    Mandar los resultados al maestro
23:  end if
24: end while
25:   Avisar al maestro de la finalización
26: end if

```

Capítulo 5

Resultados

Los resultados obtenidos son la consecuencia directa de la aplicación de los métodos del capítulo 4. El análisis inicial lleva a cabo contrastes de hipótesis entre los grupos en los que se clasifica la población para evaluar la capacidad discriminativa de los marcadores. El análisis masivo determina las combinaciones óptimas de marcadores que maximizan las métricas propuestas según el modelo empleado. La elección del mejor modelo viene de la realización de las siguientes pruebas: robustez ante valores faltantes, análisis en función del número de visitas, pronóstico y reconstrucción de la historia natural. En la primera se evalúan los modelos tras introducir distintos porcentajes de valores faltantes. La segunda analiza la influencia en las métricas de las visitas disponibles por paciente. La tercera realiza estimaciones a futuro, suponiendo desconocidas las próximas n visitas. La última determina el tiempo de construcción de todos los sujetos para ordenarlos en la escala temporal y determinar las *pathological scores*. Se efectúan dos comparaciones: entre los métodos paramétricos, y entre el mejor de ellos y las redes neuronales recurrentes.

5.1. Caracterización de la población de estudio

La población ADNI se clasifica según su diagnóstico al inicio del estudio según tengan medidas de biomarcadores NM (tabla 5.1) o NM y CSF (tabla 5.2). Además, se distingue entre los pacientes estables y progresivos con solo medidas NM (tabla 5.3) o NM y CSF (tabla 5.4). En las tablas mencionadas se muestran la media, varianza, niveles máximos y mínimos de cada biomarcador para cada uno de los grupos; así como el análisis ANOVA o Chi Cuadrado con prueba post-hoc (método de Tukey).

Como puede observarse, la población MCI es la más numerosa, mientras que los SMC representan el grupo más pequeño. El número de visitas por sujeto es considerablemente menor en la población NM+CSF que en la NM. Este resultado es coherente puesto que el coste (temporal y de recursos) de las pruebas CSF es mayor que las NM.

El 15 % de la población cognitivamente normal (CU y SMC) progresa a deterioro cognitivo leve (MCI). Un 81 % de la población CU es estable (sCU) frente a un 19 % progresiva (pCU). En la población SMC, los estables (sSMC) ascienden a un 90 % y los progresivos (pSMC) descienden a un 10 %. La población pCU presenta un tiempo de seguimiento medio de unos 8 años, mientras que los pSMC de unos 4.5 años. Por lo tanto, los resultados obtenidos para sujetos pCU son más fiables que los pSMC (es más probable que un sujeto sea progresivo si ha sido estudiado durante más tiempo). Entre los sujetos progresivos, el 37 % y el 61.5 % de los pCU y pSMC, respectivamente, convierten a menos de 4 años (con tiempos de seguimiento medios de aproximadamente 4 y 3 años, respectivamente).

La presencia de *APOE* $\epsilon 4$ aumenta cuanto mayor es el deterioro (alrededor del 27.5 % en sujetos CU y del 67.5 % en AD). Esto también se puede apreciar en los resultados de las pruebas NM. Por ejemplo, cuanto mayor es el deterioro cognitivo menor es la puntuación en MMSE (mejor funcionamiento cognitivo cuanto mayor sea la puntuación) y mayor es en ADAS13 (mayor deterioro a mayor puntuación). En los sujetos con medidas CSF, la cantidad de $A\beta$, pTau y Tau es mayor en sujetos AD y MCI que en CU. Por otro lado, se observa que los tiempos de conversión son mayores en sujetos progresivos. Los sujetos SMC convierten antes (entre 55 y 70 semanas) que los CU (≈ 100 semanas).

La tabla 5.5 presenta las características demográficas y clínicas de la población usada en el entrenamiento de los algoritmos. Los sujetos se dividen en sCU, pCU y sMCI. Aquellos pacientes con una sola visita han sido eliminados.

Group	CN	SMC	MCI	AD	Post-Hoc
Subjects	542 (22.3 %)	353 (14.5 %)	1112 (45.8 %)	411 (16.9 %)	
Visits	3139 (27.4 %)	990 (8.7 %)	6067 (53.0 %)	1244 (10.9 %)	
Female	287 (53.0 %)	220 (62.3 %)	461 (41.5 %)	180 (43.8 %)	b,d,e
APOE4	142 (n=514 27.6 %)	103 (n=285 36.1 %)	507 (n=1021 49.7 %)	261 (n=389 67.1 %)	b,c,d,e,f
Age	73.3 (6.4) (55.1, 89.6)	70.6 (6.8) (50.4, 90.3)	72.8 (7.6) (54.4, 91.4)	74.7 (7.9) (55.1, 90.9)	a,d,e,f
Education	16.4 (2.6) (6.0, 20.0)	16.7 (2.3) (8.0, 20.0)	16.0 (2.8) (4.0, 20.0)	15.2 (2.9) (4.0, 20.0)	c,d,e,f
MMSE	29.1 (1.1) (24.0, 30.0)	29.1 (1.2) (24.0, 30.0)	27.6 (1.9) (19.0, 30.0)	23.1 (2.2) (16.0, 30.0)	b,c,d,e,f
CDRSB	0.03 (0.12) (0.00, 1.00)	0.05 (0.16) (0.00, 1.00)	1.52 (0.93) (0.00, 6.00)	4.43 (1.69) (1.00, 10.00)	b,c,d,e,f
ADAS13	9.0 (4.4) (0.0, 24.7)	8.6 (4.3) (0.7, 26.3)	16.4 (6.8) (0.7, 39.7)	30.0 (8.0) (12.7, 54.7)	b,c,d,e,f
FAQ	0.1 (0.5) (0.0, 6.0)	0.3 (1.1) (0.0, 13.0)	3.2 (4.1) (0.0, 24.0)	13.3 (6.9) (0.0, 30.0)	b,c,d,e,f
LDELTOTAL	13.3 (3.3) (5.0, 22.0)	12.7 (3.5) (3.0, 23.0)	5.9 (3.5) (0.0, 20.0)	1.4 (2.0) (0.0, 11.0)	b,c,d,e,f
TRABSCORE	82.2 (41.5) (6.0, 300.0)	79.8 (38.0) (31.0, 300.0)	115.8 (66.0) (0.0, 300.0)	194.9 (87.8) (35.0, 300.0)	b,c,d,e,f
PACC	0.0 (2.7) (-12.6, 6.3)	-0.2 (2.8) (-9.3, 5.5)	-6.2 (4.0) (-22.0, 2.7)	-15.1 (3.5) (-23.6, -3.0)	b,c,d,e,f
RAVLT-Im	45.4 (10.1) (16.0, 71.0)	46.0 (10.2) (18.0, 70.0)	34.4 (10.5) (7.0, 68.0)	22.9 (7.5) (0.0, 47.0)	b,c,d,e,f
RAVLT-Lear	6.0 (2.4) (-2.0, 11.0)	6.3 (2.3) (0.0, 12.0)	4.1 (2.6) (-4.0, 12.0)	1.9 (1.8) (-4.0, 8.0)	b,c,d,e,f
RAVLT-perFor	34.3 (27.5) (-100.0, 100.0)	35.4 (28.0) (-37.5, 100.0)	60.2 (34.3) (-316.7, 100.0)	89.5 (21.8) (-50.0, 100.0)	b,c,d,e,f

Tabla 5.1: Características demográficas y clínicas iniciales de la población. Sujetos CN, SMC, MCI y AD con marcadores NM. Tests ANOVA (todas) y Chi cuadrado (APOE4 y Female) con $p < 0.01$ y prueba Tukey post-hoc: CN-SMC (a), CN-MCI(b), CN-AD(c), SMC-MCI(d), SMC-AD(e), MCI-AD(f).

Group	CN	SMC	MCI	AD	Post-Hoc
Subjects	272 (21.7%)	95 (7.6%)	619 (49.3%)	229 (18.2%)	
Visits	650 (27.5%)	140 (5.9%)	1233 (52.1%)	344 (14.5%)	
Female	137 (50.4%)	56 (58.9%)	255 (41.2%)	95 (41.5%)	d
APOE4	70 (n=272 25.7%)	33 (n=95 34.7%)	305 (n=619 49.3%)	156 (n=229 68.1%)	b,c,d,e,f
Age	74.4 (6.0) (56.2, 89.6)	72.0 (5.4) (59.7, 85.3)	72.3 (7.5) (54.4, 91.4)	74.6 (8.2) (55.6, 90.3)	b,f
Education	16.2 (2.7) (6.0, 20.0)	16.7 (2.5) (8.0, 20.0)	16.1 (2.8) (6.0, 20.0)	15.4 (2.9) (4.0, 20.0)	c,e
MMSE	29.1 (1.1) (24.0, 30.0)	29.0 (1.2) (24.0, 30.0)	27.7 (1.8) (23.0, 30.0)	23.3 (2.0) (19.0, 26.0)	b,c,d,e,f
CDRSB	0.03 (0.12) (0.00, 1.00)	0.08 (0.18) (0.00, 0.50)	1.48 (0.88) (0.50, 5.50)	4.46 (1.65) (1.00, 10.00)	b,c,d,e,f
ADAS13	9.4 (4.4) (0.0, 24.0)	8.6 (4.0) (2.0, 20.0)	16.1 (6.9) (2.0, 38.0)	29.9 (8.1) (14.0, 54.7)	b,c,d,e,f
FAQ	0.2 (0.7) (0.0, 6.0)	0.6 (1.8) (0.0, 13.0)	3.1 (4.0) (0.0, 22.0)	13.2 (7.0) (0.0, 28.0)	b,c,d,e,f
LDELTOTAL	13.2 (3.3) (5.0, 22.0)	13.2 (3.3) (5.0, 23.0)	6.1 (3.4) (0.0, 15.0)	1.4 (1.8) (0.0, 8.0)	b,c,d,e,f
TRABSCORE	84.2 (43.0) (36.0, 300.0)	85.7 (40.3) (32.0, 300.0)	113.9 (64.0) (33.0, 300.0)	196.4 (86.8) (42.0, 300.0)	b,c,d,e,f
PACC	-0.1 (2.6) (-9.6, 5.4)	-0.1 (2.8) (-6.6, 5.5)	-6.0 (4.0) (-16.6, 2.2)	-14.9 (3.3) (-22.4, -7.1)	b,c,d,e,f
RAVLT-Im	44.8 (9.9) (18.0, 71.0)	46.4 (9.9) (27.0, 70.0)	34.7 (10.7) (11.0, 68.0)	22.9 (7.2) (1.0, 46.0)	b,c,d,e,f
RAVLT-Lear	5.8 (2.3) (0.0, 11.0)	6.2 (2.2) (0.0, 10.0)	4.2 (2.6) (-2.0, 12.0)	1.8 (1.7) (-2.0, 7.0)	b,c,d,e,f
RAVLT-perFor	34.8 (28.4) (-100.0, 100.0)	36.9 (28.7) (-33.3, 100.0)	60.2 (32.1) (-36.4, 100.0)	88.0 (22.6) (0.0, 100.0)	b,c,d,e,f
ABETA	1191.0 (451.3) (200.0, 1700.0)	1245.5 (421.5) (391.2, 1700.0)	961.4 (437.2) (210.9, 1700.0)	669.1 (315.7) (212.3, 1700.0)	b,c,d,e,f
TAU	236.7 (90.0) (80.0, 590.1)	239.3 (93.2) (88.7, 501.3)	286.9 (134.5) (97.9, 1300.0)	367.0 (145.8) (80.0, 851.6)	b,c,d,e,f
pTAU	21.8 (9.2) (8.0, 60.0)	21.8 (9.6) (8.5, 53.3)	27.8 (15.0) (8.2, 120.0)	36.6 (15.8) (8.0, 94.9)	b,c,d,e,f
TAU/ABETA	0.24 (0.18) (0.09, 1.61)	0.22 (0.14) (0.09, 0.76)	0.39 (0.28) (0.07, 2.13)	0.63 (0.31) (0.10, 2.40)	b,c,d,e,f
pTAU/ABETA	0.02 (0.02) (0.01, 0.18)	0.02 (0.01) (0.01, 0.08)	0.04 (0.03) (0.01, 0.24)	0.06 (0.03) (0.01, 0.25)	b,c,d,e,f

Tabla 5.2: Características demográficas y clínicas iniciales de la población. Sujetos CN, SMC, MCI y AD con marcadores NM y CSF. Tests ANOVA (todas) y Chi cuadrado (APOE4 y Female) con $p < 0.01$ y prueba Tukey post-hoc: CN-SMC (a), CN-MCI(b), CN-AD(c), SMC-MCI(d), SMC-AD(e), MCI-AD(f).

Group	sCU	pCU	sSMC	pSMC	Post-Hoc
Subjects	440 (49.3 %)	100 (11.2 %)	319 (35.7 %)	34 (3.8 %)	
Visits	2270 (55.0 %)	869 (21.0 %)	839 (20.3 %)	151 (3.7 %)	
Censure time	56.9 (47.2) (0.0, 186.0)	96.1 (47.8) (6.0, 204.0)	31.1 (30.4) (0.0, 114.0)	55.8 (32.2) (12.0, 114.0)	
Convert time	0.0 (0.0) (0.0, 0.0)	60.4 (37.1) (6.0, 186.0)	0.0 (0.0) (0.0, 0.0)	39.9 (27.5) (6.0, 102.0)	
Female	238 (54.1 %)	47 (47.0 %)	203 (63.6 %)	17 (50.0 %)	
APOE4	107 (n=413 25.9 %)	35 (n=100 35.0 %)	91 (n=252 36.1 %)	12 (n=33 36.4 %)	
Age	72.8 (6.5) (55.1, 89.6)	75.9 (5.2) (63.2, 89.0)	70.2 (6.8) (50.4, 90.3)	73.9 (6.0) (65.1, 90.1)	a,b,d,f
Education	16.5 (2.6) (6.0, 20.0)	16.2 (2.6) (8.0, 20.0)	16.7 (2.3) (8.0, 20.0)	16.2 (2.4) (12.0, 20.0)	
MMSE	29.1 (1.2) (24.0, 30.0)	29.0 (1.1) (24.0, 30.0)	29.1 (1.1) (24.0, 30.0)	29.0 (1.4) (24.0, 30.0)	
CDRSB	0.03 (0.12) (0.00, 1.00)	0.04 (0.14) (0.00, 0.50)	0.04 (0.14) (0.00, 0.50)	0.13 (0.26) (0.00, 1.00)	c,e,f
ADAS13	8.6 (4.2) (0.0, 24.7)	10.8 (4.7) (1.0, 24.0)	8.3 (4.2) (0.7, 21.7)	11.3 (4.7) (4.0, 26.3)	a,c,d,f
FAQ	0.1 (0.4) (0.0, 5.0)	0.3 (0.9) (0.0, 6.0)	0.2 (0.9) (0.0, 8.0)	0.7 (2.3) (0.0, 13.0)	c
LDELTOTAL	13.5 (3.3) (5.0, 22.0)	12.3 (3.3) (6.0, 22.0)	12.8 (3.4) (3.0, 23.0)	11.5 (3.7) (3.0, 20.0)	c
TRABSCORE	81.0 (43.0) (6.0, 300.0)	87.8 (34.4) (39.0, 244.0)	77.4 (34.5) (31.0, 300.0)	102.5 (57.9) (39.0, 300.0)	f
PACC	0.2 (2.7) (-12.6, 6.3)	-0.9 (2.9) (-9.6, 4.7)	-0.1 (2.7) (-8.4, 5.5)	-1.6 (3.2) (-9.3, 4.3)	a,c
RAVLT-Im	46.3 (10.1) (16.0, 71.0)	41.6 (9.2) (22.0, 67.0)	46.8 (10.1) (18.0, 70.0)	39.0 (7.7) (27.0, 54.0)	a,c,d,f
RAVLT-Lear	6.1 (2.3) (0.0, 11.0)	5.3 (2.6) (-2.0, 11.0)	6.3 (2.3) (1.0, 12.0)	5.6 (2.4) (0.0, 10.0)	a,d
RAVLT-perFor	32.9 (27.0) (-100.0, 100.0)	39.7 (28.5) (-20.0, 100.0)	34.4 (27.8) (-37.5, 100.0)	45.3 (29.1) (0.0, 100.0)	

Tabla 5.3: Características demográficas y clínicas iniciales de la población. Sujetos sCU, pCU, sSMC y pSMC con marcadores NM. Tests ANOVA (todas) y Chi cuadrado (APOE4 y Female) con $p < 0.01$ y prueba Tukey post-hoc: sCU-pCU (a), sCU-sSMC(b), sCU-pSMC(c), pCU-sSMC(d), pCU-pSMC(e), sSMC-pSMC(f).

Group	sCU	pCU	sSMC	pSMC	Post-Hoc
Subjects	209 (55.1 %)	63 (16.6 %)	79 (20.8 %)	16 (4.2 %)	
Visits	451 (57.1 %)	199 (25.2 %)	115 (14.6 %)	25 (3.2 %)	
Censure time	69.7 (44.1) (0.0, 186.0)	100.9 (48.8) (6.0, 204.0)	57.6 (36.5) (0.0, 114.0)	68.2 (36.2) (12.0, 114.0)	
Convert time	0.0 (0.0) (0.0, 0.0)	60.3 (37.7) (6.0, 180.0)	0.0 (0.0) (0.0, 0.0)	46.5 (34.5) (6.0, 102.0)	
Female	109 (52.2 %)	28 (44.4 %)	49 (62.0 %)	7 (43.8 %)	
APOE4	51 (n=209 24.4 %)	19 (n=63 30.2 %)	25 (n=79 31.6 %)	8 (n=16 50.0 %)	
Age	73.8 (6.1) (56.2, 89.6)	76.3 (5.3) (63.2, 89.0)	71.7 (5.5) (59.7, 85.3)	73.8 (4.5) (66.7, 80.4)	d
Education	16.3 (2.6) (6.0, 20.0)	16.0 (2.8) (8.0, 20.0)	16.8 (2.5) (8.0, 20.0)	16.1 (2.4) (12.0, 20.0)	
MMSE	29.1 (1.1) (24.0, 30.0)	28.9 (1.2) (24.0, 30.0)	29.0 (1.2) (25.0, 30.0)	28.9 (1.5) (24.0, 30.0)	
CDRSB	0.03 (0.12) (0.00, 1.00)	0.03 (0.12) (0.00, 0.50)	0.06 (0.17) (0.00, 0.50)	0.16 (0.24) (0.00, 0.50)	c,e
ADAS13	8.9 (4.2) (0.0, 23.0)	11.1 (4.7) (2.0, 24.0)	8.2 (4.0) (2.0, 20.0)	10.4 (3.9) (4.0, 16.0)	a,d
FAQ	0.1 (0.5) (0.0, 5.0)	0.3 (1.0) (0.0, 6.0)	0.5 (1.3) (0.0, 7.0)	1.4 (3.2) (0.0, 13.0)	c,e
LDELTOTAL	13.5 (3.1) (5.0, 21.0)	12.0 (3.5) (6.0, 22.0)	13.3 (3.2) (5.0, 23.0)	12.4 (3.3) (7.0, 20.0)	
TRABSCORE	83.0 (44.8) (36.0, 300.0)	88.3 (36.3) (46.0, 244.0)	84.1 (41.4) (32.0, 300.0)	94.0 (34.5) (48.0, 173.0)	
PACC	0.2 (2.4) (-8.9, 5.4)	-1.1 (2.9) (-9.6, 4.7)	-0.0 (2.9) (-6.5, 5.5)	-0.8 (2.7) (-6.6, 4.3)	a
RAVLT-Im	46.1 (9.9) (18.0, 71.0)	40.7 (8.9) (22.0, 64.0)	48.0 (9.4) (28.0, 70.0)	38.4 (8.1) (27.0, 54.0)	a,d,f
RAVLT-Lear	5.9 (2.2) (0.0, 11.0)	5.5 (2.5) (0.0, 11.0)	6.4 (2.1) (1.0, 10.0)	5.2 (2.6) (0.0, 10.0)	
RAVLT-perFor	33.0 (28.4) (-100.0, 100.0)	40.9 (28.1) (-20.0, 100.0)	36.1 (28.1) (-33.3, 100.0)	40.4 (32.4) (0.0, 100.0)	
ABETA	1228.2 (438.8) (203.0, 1700.0)	1067.7 (473.4) (200.0, 1700.0)	1298.6 (399.8) (416.8, 1700.0)	983.6 (440.9) (391.2, 1700.0)	
TAU	227.7 (87.3) (80.0, 590.1)	266.7 (93.0) (113.2, 462.8)	236.3 (89.1) (88.7, 501.3)	254.2 (113.2) (113.4, 492.1)	
pTAU	20.9 (8.9) (8.0, 60.0)	24.9 (9.4) (9.9, 48.3)	21.2 (8.6) (8.5, 53.3)	24.7 (13.3) (10.3, 52.3)	
TAU/ABETA	0.22 (0.15) (0.09, 1.03)	0.32 (0.24) (0.10, 1.61)	0.20 (0.11) (0.09, 0.68)	0.32 (0.20) (0.10, 0.76)	a,d
pTAU/ABETA	0.02 (0.02) (0.01, 0.12)	0.03 (0.03) (0.01, 0.18)	0.02 (0.01) (0.01, 0.07)	0.03 (0.02) (0.01, 0.08)	a,d

Tabla 5.4: Características demográficas y clínicas iniciales de la población. Sujetos sCU, pCU, sSMC y pSMC con marcadores NM y CSF. Tests ANOVA (todas) y Chi cuadrado (APOE4 y Female) con $p < 0.01$ y prueba Tukey post-hoc: sCU-pCU (a), sCU-sSMC(b), sCU-pSMC(c), pCU-sSMC(d), pCU-pSMC(e), sSMC-pSMC(f).

Group	sCU	pCU	sMCI	Post-Hoc
Subjects	536 (46.1 %)	114 (9.8 %)	513 (44.1 %)	
Visits	2678 (42.8 %)	795 (12.7 %)	2783 (44.5 %)	
Female	298 (55.6 %)	52 (45.6 %)	203 (39.6 %)	a,b,c
APOE4	155 (n=531 29.2%)	41 (n=114 36.0%)	216 (n=506 42.7 %)	a,b,c
Age	73.0 (6.2) (56.2, 90.3)	75.5 (5.7) (63.2, 90.1)	73.0 (7.6) (55.0, 91.4)	a,b,c
Education	16.6 (2.6) (6.0, 20.0)	16.1 (2.6) (8.0, 20.0)	15.8 (2.9) (4.0, 20.0)	a,b
MMSE	29.1 (1.1) (24.0, 30.0)	29.0 (1.2) (24.0, 30.0)	27.9 (1.8) (19.0, 30.0)	b,c
CDRSB	0.03 (0.13) (0.00, 1.00)	0.06 (0.16) (0.00, 0.50)	1.30 (0.76) (0.00, 5.50)	b,c
ADAS13	9.6 (4.3) (0.0, 23.0)	11.5 (5.0) (1.0, 29.3)	15.2 (5.9) (3.0, 37.0)	a,b,c
FAQ	0.2 (0.7) (0.0, 7.0)	0.4 (1.4) (0.0, 13.0)	2.1 (3.2) (0.0, 21.0)	b,c
LDELTOTAL	13.4 (3.2) (5.0, 23.0)	11.9 (3.4) (3.0, 22.0)	6.6 (3.2) (0.0, 20.0)	a,b,c
TRABSCOR	81.1 (42.8) (6.0, 300.0)	93.2 (42.6) (46.0, 300.0)	106.1 (56.5) (6.0, 300.0)	a,b,c
PACC	0.1 (2.6) (-9.2, 6.3)	-1.3 (2.9) (-9.9, 4.7)	-5.4 (3.6) (-21.1, 3.4)	a,b,c
RAVLT-Im	46.4 (9.8) (18.0, 71.0)	40.6 (9.1) (22.0, 67.0)	36.6 (10.7) (13.0, 68.0)	a,b,c
RAVLT-Lear	6.1 (2.3) (0.0, 12.0)	5.2 (2.6) (-2.0, 11.0)	4.6 (2.5) (-4.0, 11.0)	a,b,c
RAVLT-perFor	32.3 (29.0) (-218.2, 100.0)	44.5 (29.3) (-20.0, 100.0)	53.4 (31.3) (-25.0, 100.0)	a,b,c

Tabla 5.5: Características demográficas y clínicas de la población de entrenamiento. Sujetos sCU, pCU, sMCI con marcadores NM. Tests ANOVA (todas) y Chi cuadrado (APOE4 y Female) con $p < 0.01$ y prueba Tukey post-hoc: sCU-pCU (a), sCU-sMCI (b), pCU-sMCI (c).

5.2. Clasificación de la población

Los sujetos sCU, pCU, sSMC y pSMC han sido clasificados según su tiempo de conversión (progresivos) o censura (estables). Aquellos que disponen de medidas CSF, se consideran positivos en patología amiloide si el ratio $\frac{p^{TAU}}{A\beta} > 0,028$. El script `histogram.ipynb` de la carpeta `/demo/clasificacion_poblacion` se encarga de generar los histogramas con dicha información (figuras 5.1, 5.2, 5.3, 5.4).

Se observa que la número de pacientes desciende cuanto mayor es el tiempo de censura. Más de la mitad de la población sCU y sSMC con solo marcadores NM tiene tiempos de censura por debajo de los 4 años. Aquellos con medidas CSF tienen tiempos de censura más largos. La cantidad de sujetos estables con patología amiloide es baja y aproximadamente constante con el tiempo de censura. El tiempo de conversión en sujetos CU es mayor que en SMC. Entre estos últimos, 15 de 22 convirtieron antes de 4 años. La prevalencia de patología amiloide es más acusada en los conversores rápidos. La proporción de sujetos progresivos con dicha patología es más significativa que en los estables.

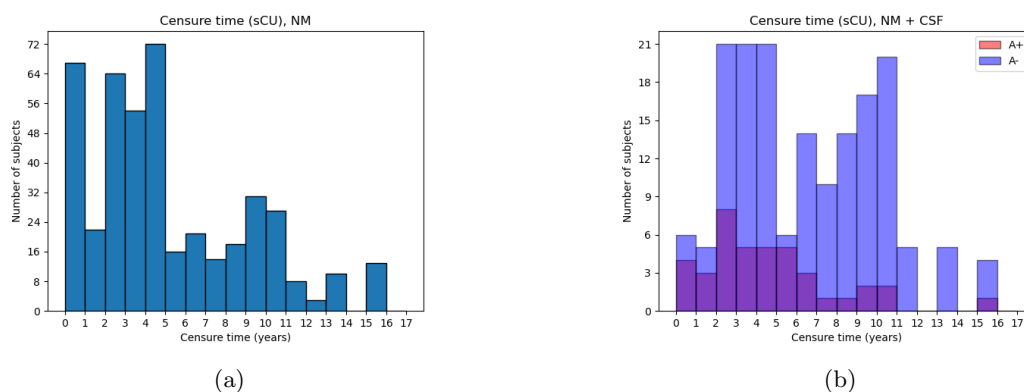


Figura 5.1: Histogramas con los tiempos de censura de sujetos sCU con medidas NM (a) y NM + CSF (b)

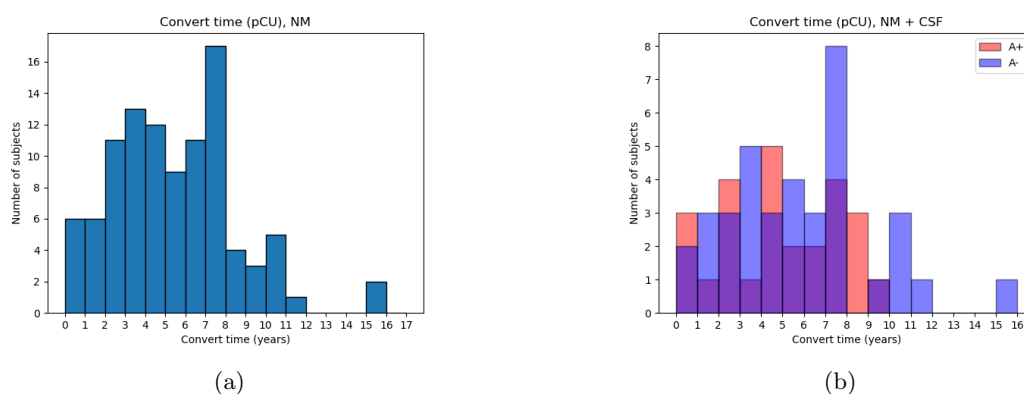


Figura 5.2: Histogramas con los tiempos de conversión de sujetos pCU con medidas NM (a) y NM + CSF (b)

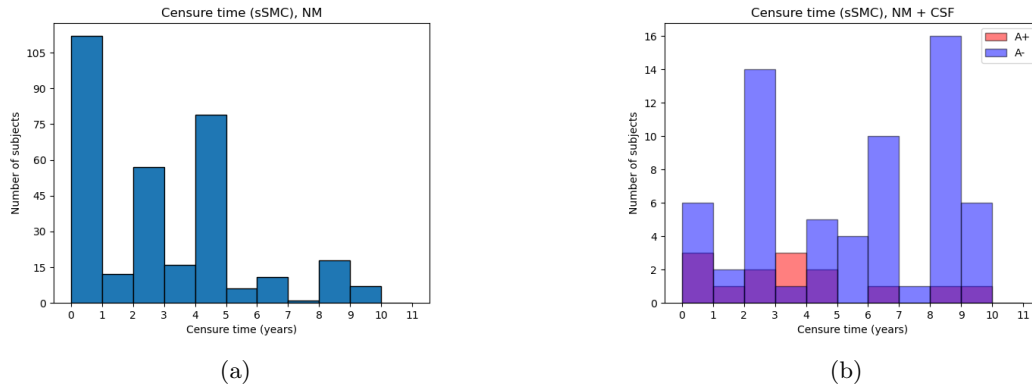


Figura 5.3: Histogramas con los tiempos de censura de sujetos sSMC con medidas NM (a) y NM + CSF (b)

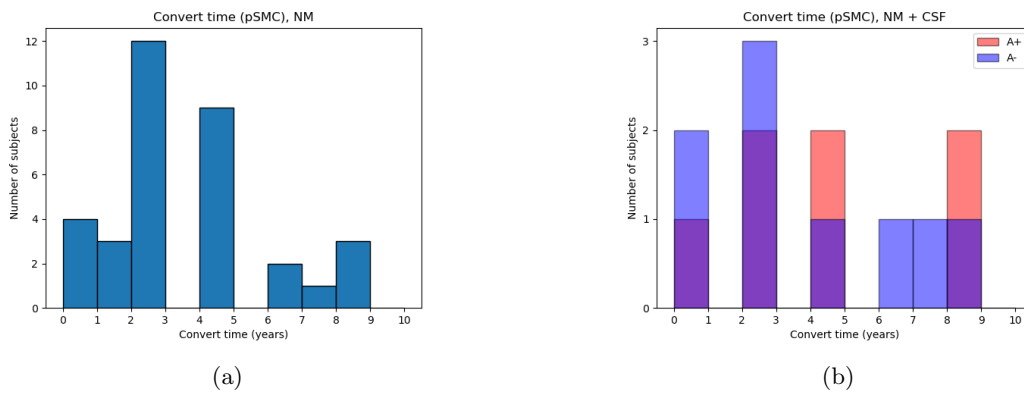


Figura 5.4: Histogramas con los tiempos de conversión de sujetos pSMC con medidas NM (a) y NM + CSF (b)

5.3. Análisis masivo

Los vectores de características se construyen con combinaciones de hasta 5 *tests* neuropsicológicos. Para elegir el mejor vector, las medidas de calidad prioritarias son la detección de sujetos pCU y la correlación con la reserva. Se consideran mejores los vectores de menor dimensión, ya que implican realizar un menor número de *tests*, con su consiguiente reducción de costes y tiempo dedicado. Los valores de *mPACCdigit* y *mPACCtrailsB* resultan de un conglomerado de pruebas, por lo que otros marcadores son preferentes.

Las tablas 5.6 y 5.8 resumen los resultados de los 5 mejores vectores de características para Leaspy y GRACE, respectivamente. La misma información se resumen en las tablas 5.7 y 5.9 para vectores de dimensión reducida (2 y 3 características) para Leaspy y GRACE, respectivamente. El ordenamiento se ha realizado en base a la media geométrica entre los sujetos de *train* y *test* del producto detección de pCU por correlación con la reserva.

Las tablas 5.26 y 5.15 exponen los valores del MAE para las combinaciones de vectores de Leaspy recogidas en las tablas 5.6 y 5.7. Las tablas 5.27 y 5.17 exponen lo mismo para el caso de GRACE.

5.3.1. Leaspy

El mejor vector en el caso de Leaspy es el formado por:

{ RAVLT Forgetting, MMSE, CDRSB, mPACCdigit, mPACCtrailsB }

En el caso de los vectores de dimensión reducida:

{ ADAS13, CDRSB }

Las puntuaciones en los sujetos de *test* para estos vectores son, respectivamente, 80.38 % y 81.15 % en detección de pCU, y 0.8158 y 0.7787 en correlación con la reserva.

A pesar de que la correlación con la reserva sea un 4 % más baja, la combinación **{ ADAS13, CDRSB }** implica la realización de 3 pruebas menos, entre las que no se incluyen *mPACCdigit* y *mPACCtrailsB*. Aunque la detección de pCU mejore en esta combinación en un 1 %, la desviación típica es aproximadamente un 4 % superior.

5.3.2. GRACE

El mejor vector en el caso de GRACE es el formado por:

{ RAVLT Forgetting, FAQ, MMSE, CDRSB }

En el caso de los vectores de dimensión reducida:

{ RAVLT Forgetting, MMSE, CDRSB }

Sin embargo, se tienen resultados similares con una característica menos con:

{ MMSE, CDRSB }

Las puntuaciones en los sujetos de *test* para estos vectores son, respectivamente, 64.00 %, 54.00 % y 57.60 % en detección de pCU, y 0.6212, 0.6380 y 0.5434 en correlación con la reserva.

Atendiendo al criterio, la dimensión del vector de características es más relevante en GRACE que en Leaspy. Mientras que la diferencia entre los vectores completos y los de dimensión reducida es menor de 2 puntos en Leaspy, asciende a más de 8 puntos en GRACE. La utilización de menos marcadores no es evidente en el caso de GRACE.

5.3.3. RPDPM

El análisis aplicado a *GRACE* y *Leaspy* se realiza sobre *RPDPM*. Las tablas 5.10 y 5.11 resumen los resultados para los mejores vectores de características completos y de dimensión reducida, respectivamente. Los valores del MAE para estos vectores se presentan en las tablas 5.28 y 5.19.

El mejor vector de características es:

{ RAVLT Forgetting, MMSE, CDRSB, mPACCtrailsB }

En el caso de los de dimensión reducida:

{ *MMSE*, *CDRSB*, *mPACCtrailsB* }

Los valores para la población de *test* de correlación con la reserva son 0.8817 y 0.8516, y de detección de pCU son 81.36 % y 76.82 %, respectivamente. La correlación con la reserva y la detección de pCU disminuyen un 3 % y 5 %.

Ambas combinaciones implican la realización de *mPACCtrailsB*, por lo que la elección de { *MMSE*, *CDRSB*, *LDELTOTAL* }¹ o { *MMSE*, *CDRSB* } es más interesante. Entre estas dos, la última supone la realización de una prueba menos y presenta unos resultados parecidos. En este caso, se tienen valores de 0.8798 y 76.82 % en la correlación con la reserva y la detección de pCU.

La reducción de la dimensión no influye de manera significativa en la determinación del mejor vector, ya que 3 de los 5 mejores vectores están compuestos por 3 o menos marcadores. La diferencia en el criterio no es superior a 2 puntos al reducir la dimensión.

En todas las combinaciones se aprecia un ligero descenso de los resultados en la población de entrenamiento (entre un 3 % y un 6 % en la correlación y entre un 5 % y un 10 % en la detección de pCU). Una posible causa podría ser la separación de la población en *train* y *test*. Esto mismo ocurre en Leaspy, pero siendo la población de *test* la que empeora sus resultados.

Atendiendo al criterio, Leaspy es el algoritmo que mejores resultados obtiene, con una distancia alrededor de 5 puntos con RPDPM. Los vectores de menor dimensión también presentan una puntuación superior en Leaspy que en RPDPM. Claramente, GRACE es el que peor desempeño tiene de todos.

En cuanto a la correlación con la reserva, RPDPM supera a Leaspy en *test*, mientras que lo contrario ocurre en *train*. Esto podría ser debido a que los conjuntos de entrenamiento y *test* son diferentes para cada modelo.

Los resultados en detección de pCU de Leaspy son mejores tanto en *train* como en *test*.

GRACE presenta los valores más reducidos de MAE, seguido por Leaspy y RPDPM, en dicho orden. La diferencia en esta medida es sutil entre Leaspy y RPDPM, mientras que GRACE es claramente superior. Esto sugiere que los algoritmos con peor desempeño temporal, resultan más eficientes en la estimación espacial de las trayectorias. Cabe decir que el MAE en CDRSB y FAQ en RPDPM es similar al de GRACE.

Al no considerar el MAE una medida significativa para determinar la calidad del algoritmo, puede afirmarse que Leaspy es superior a RPDPM, y este es superior a GRACE.

5.3.4. LSTM

El análisis masivo se realiza sobre *LSTM*. La longitud de secuencia se establece en 4, con la consiguiente eliminación de los sujetos con menor cantidad de visitas. El intervalo entre puntos temporales se fija en 6 meses, y se interpolan los valores faltantes. La arquitectura de la red presenta 64 neuronas con una capa oculta. A la salida de la capa oculta se aplica una función lineal para obtener las estimaciones de los marcadores. La salida

¹Nótese que ADNI realiza el diagnóstico según la terna *MMSE*, *CDRSB*, *LDELTOTAL*.

binaria del diagnóstico se consigue introduciendo el resultado anterior en una sigmoide. La función de pérdida pondera igualmente las contribuciones de MAE y BCE. Las tablas 5.12 y 5.13 resumen los resultados para los mejores vectores de características completos y de dimensión reducida, respectivamente. Los valores del MAE para estos vectores se presentan en las tablas 5.29 y 5.21.

El mejor vector de características es:

$$\{ \mathbf{RAVLT\ Immediate}, \mathbf{RAVLT\ Learning}, \mathbf{CDRSB}, \mathbf{mPACCdigit}, \mathbf{mPACCtrailsB} \}$$

En el caso de los de dimensión reducida:

$$\{ \mathbf{CDRSB}, \mathbf{mPACCdigit}, \mathbf{mPACCtrailsB} \}$$

Los valores para la población de *test* de correlación con la reserva son 0.7197 y 0.7193, y de detección de pCU son 100.00 % y 100.00 %, respectivamente. No hay diferencia apreciable entre la correlación con la reserva y la detección de pCU.

Ambas combinaciones implican la realización de *mPACCtrailsB* y *mPACCdigit*, por lo que la elección de $\{ \mathbf{RAVLT\ Learning}, \mathbf{FAQ}, \mathbf{CDRSB} \}$ resulta de mayor interés. La correlación con la reserva y la detección de pCU se sitúan en 0.7179 y 98.00 %, respectivamente.

La reducción de dimensión afecta levemente en el rendimiento de los modelos. Las mejores combinaciones no incluyen menos de 5 marcadores, mientras que las de dimensión reducida no bajan de 3. Esto podría ser síntoma de un problema de sobreajuste. Debido a que las redes neuronales tienen miles de parámetros a ajustar, reducir la dimensión implica la memorización de datos². En cualquier caso, las métricas se mantienen estables, con una diferencia no superior a 3 puntos en el criterio. La detección de pCU es próxima al 100 % en *test*. No obstante, el número de muestras de dicha población es escaso debido a la eliminación de sujetos de menos de 4 visitas. En *train* la detección de pCU baja, mientras que la correlación con la reserva sube. En general, las puntuaciones en ambos conjuntos son equilibradas.

Atendiendo al criterio, LSTM se encuentra a medio camino entre Leaspy y RPDPM. En la detección de pCU supera a los modelos paramétricos, aunque debe tenerse en cuenta que estos incluyen sujetos con 2 y 3 visitas. La correlación con la reserva difícilmente se eleva por encima de 0.8000. Como es lógico, al detectar un mayor porcentaje de sujetos progresivos disminuye el rendimiento al estimar el tiempo de conversión.

El ajuste espacial en términos de MAE supera con creces a los métodos paramétricos. A excepción de RAVLT Learning y mPACCdigit, se impone sobre el resto de marcadores. Especialmente en RAVLT Perc Forgetting y TRABSCOR presenta unas mejoras significativas.

Sin tener en cuenta el MAE, el desempeño es similar al de Leaspy y RPDPM.

²La cantidad de datos por parámetro disminuye. En vez de aprender relaciones generalizables, la red tiende a memorizar situaciones específicas.

		RF, M, C, P, PB	RF, C, P, PB	RP, C, P, T	M, C, L, P, PB	M, C, P, PB
Test	Corr. edad	0.9917 (1.12e-03)	0.9906 (1.24e-03)	0.9893 (1.24e-03)	0.9912 (1.27e-03)	0.9905 (1.07e-03)
	Corr. reserva	0.8158 (2.33e-02)	0.8021 (1.95e-02)	0.8009 (1.10e-02)	0.8080 (2.08e-02)	0.8007 (1.45e-02)
	% sCU	78.13 (2.14)	76.64 (0.42)	77.10 (1.27)	77.20 (2.30)	76.54 (1.28)
	% sMCI	93.90 (0.30)	94.90 (0.70)	95.40 (0.49)	95.10 (0.94)	94.70 (0.90)
	% pCU	80.38 (2.07)	81.15 (1.15)	81.92 (2.46)	80.77 (4.55)	81.15 (3.19)
	AUC	0.9567 (5.82e-03)	0.9511 (3.30e-03)	0.9580 (6.03e-03)	0.9616 (7.74e-03)	0.9506 (5.53e-03)
Train	Corr. edad	0.9945 (4.58e-04)	0.9942 (4.76e-04)	0.9934 (6.26e-04)	0.9943 (1.04e-03)	0.9942 (9.98e-04)
	Corr. reserva	0.8909 (1.07e-02)	0.8866 (8.59e-03)	0.8725 (1.04e-02)	0.8865 (2.24e-02)	0.8855 (2.06e-02)
	% sCU	83.08 (1.21)	81.21 (1.54)	81.00 (1.46)	81.79 (1.81)	81.49 (1.50)
	% sMCI	95.16 (0.67)	96.08 (0.79)	96.22 (0.99)	96.51 (0.94)	96.10 (0.77)
	% pCU	82.05 (2.02)	81.93 (1.79)	82.39 (2.70)	81.25 (3.10)	81.59 (1.96)
	AUC	0.9710 (8.27e-04)	0.9707 (1.78e-03)	0.9731 (2.34e-03)	0.9720 (1.48e-03)	0.9693 (3.66e-03)
Criterio		69.23	68.76	68.67	68.56	68.52

Tabla 5.6: Mejores vectores de características para el algoritmo Leaspy. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto de detección de pCU por correlación con la reserva. RF=RAVLT_forgetting, M=MMSE, C=CDRSB, P=mPACCdigit, PB=mPACCtrailsB, RP=RAVLT_perc_forgetting, T=TRABSCOR, L=LDELTOTAL.

		A, C	A, C, PB	A, M, C	A, C, P	RF, A, C
Test	Corr. edad	0.9886 (1.50e-03)	0.9890 (1.49e-03)	0.9886 (1.22e-03)	0.9887 (1.11e-03)	0.9887 (2.02e-03)
	Corr. reserva	0.7787 (2.70e-02)	0.7846 (1.31e-02)	0.7764 (2.14e-02)	0.7892 (1.42e-02)	0.7759 (4.09e-02)
	% sCU	75.42 (1.26)	76.45 (0.56)	75.98 (0.43)	76.82 (0.70)	75.33 (0.95)
	% sMCI	94.60 (0.66)	94.40 (0.66)	94.40 (0.80)	94.40 (0.49)	95.40 (0.66)
	% pCU	81.15 (5.56)	78.08 (4.57)	79.23 (2.55)	78.85 (2.58)	79.62 (3.87)
	AUC	0.9502 (9.16e-03)	0.9564 (6.54e-03)	0.9582 (7.01e-03)	0.9567 (7.78e-03)	0.9510 (7.39e-03)
Train	Corr. edad	0.9925 (1.44e-03)	0.9937 (3.15e-04)	0.9921 (1.61e-03)	0.9936 (3.83e-04)	0.9922 (1.27e-03)
	Corr. reserva	0.8600 (2.40e-02)	0.8775 (6.20e-03)	0.8544 (2.57e-02)	0.8751 (5.50e-03)	0.8536 (1.94e-02)
	% sCU	79.30 (1.56)	81.75 (0.89)	80.26 (0.73)	82.12 (1.10)	79.42 (1.93)
	% sMCI	95.16 (1.41)	94.99 (0.93)	94.84 (1.34)	95.21 (0.92)	94.99 (1.37)
	% pCU	83.64 (6.25)	83.75 (1.14)	85.34 (3.65)	81.93 (3.54)	84.55 (6.04)
	AUC	0.9683 (3.02e-03)	0.9707 (3.51e-03)	0.9719 (2.24e-03)	0.9714 (1.51e-03)	0.9692 (3.18e-03)
Criterio		67.42	67.10	66.97	66.79	66.77

Tabla 5.7: Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo Leaspy. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva. A=ADAS13, C=CDRSB, PB=mPACctrailsB, M=MMSE, RF=RAVLT_forgetting, P=mPACcdigit.

		RF, F, M, C	RF, RP, F, M, C	A, F, M, C	RF, RL, F, M, C	A, F, M, C, P
Test	Corr. edad	0.9812 (1.09e-03)	0.9802 (9.82e-04)	0.9775 (1.82e-03)	0.9810 (1.11e-03)	0.9814 (1.02e-03)
	Corr. reserva	0.6212 (2.88e-02)	0.5565 (2.45e-02)	0.5545 (5.30e-02)	0.5622 (2.58e-02)	0.5833 (4.88e-02)
	% sCU	77.43 (2.66)	71.90 (1.12)	79.81 (2.53)	71.33 (4.39)	77.43 (1.10)
	% sMCI	74.80 (2.96)	70.20 (4.11)	75.69 (1.65)	70.00 (0.83)	73.92 (1.86)
	% pCU	64.00 (6.80)	66.00 (4.32)	67.20 (5.59)	60.40 (1.26)	58.00 (4.71)
	AUC	0.9268 (5.22e-03)	0.9122 (3.51e-03)	0.9120 (2.96e-03)	0.9136 (2.18e-03)	0.8994 (5.55e-04)
Train	Corr. edad	0.9828 (1.03e-03)	0.9816 (5.36e-04)	0.9779 (8.07e-04)	0.9796 (6.97e-04)	0.9813 (1.05e-03)
	Corr. reserva	0.6982 (1.50e-02)	0.6840 (1.26e-02)	0.6739 (1.17e-02)	0.6847 (1.11e-02)	0.7050 (1.14e-02)
	% sCU	79.00 (2.60)	77.52 (1.31)	78.10 (3.84)	77.82 (3.17)	77.38 (1.39)
	% sMCI	74.04 (1.96)	71.46 (2.40)	73.02 (1.74)	70.85 (0.95)	73.02 (1.93)
	% pCU	49.10 (3.22)	49.78 (1.68)	48.09 (4.79)	50.90 (1.50)	45.73 (2.12)
	AUC	0.9143 (4.91e-03)	0.8948 (3.18e-03)	0.9177 (2.75e-03)	0.9043 (1.77e-03)	0.9093 (5.67e-04)
Criterio		36.92	35.36	34.75	34.40	33.03

Tabla 5.8: Mejores vectores de características para el algoritmo GRACE. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva. RF=RAVLT_forgetting, F=FAQ, M=MMSE, C=CDRSB, RP=RAVLT_perc_forgetting, L=LDELTOTAL, P=mPACCdigit, RL=RAVLT_learning.

		RF, M, C	F, M, C	M, C	F, C, P	RF, A, C
Test	Corr. edad	0.9845 (1.39e-03)	0.9840 (2.32e-03)	0.9801 (1.27e-03)	0.9805 (1.49e-03)	0.9823 (1.44e-03)
	Corr. reserva	0.6380 (3.38e-02)	0.6420 (4.07e-02)	0.5634 (4.18e-02)	0.6320 (1.79e-02)	0.6307 (2.11e-02)
	% sCU	77.33 (1.84)	83.24 (2.02)	77.71 (2.77)	88.19 (1.97)	83.33 (1.63)
	% sMCI	74.61 (3.44)	76.76 (2.12)	75.69 (5.69)	84.22 (1.08)	80.00 (5.17)
	% pCU	54.00 (3.89)	46.00 (3.89)	57.60 (6.31)	45.20 (3.29)	49.20 (5.01)
	AUC	0.9206 (6.62e-03)	0.9263 (1.19e-03)	0.9225 (1.50e-03)	0.9431 (7.66e-04)	0.9260 (5.96e-03)
Train	Corr. edad	0.9802 (1.15e-03)	0.9778 (1.41e-03)	0.9805 (7.19e-04)	0.9786 (8.61e-04)	0.9775 (2.01e-03)
	Corr. reserva	0.6391 (2.46e-02)	0.6271 (6.91e-02)	0.6375 (2.22e-02)	0.6077 (2.68e-02)	0.6205 (2.03e-02)
	% sCU	81.09 (2.01)	83.67 (2.38)	82.06 (2.29)	85.66 (2.26)	81.81 (1.84)
	% sMCI	73.89 (2.09)	77.20 (1.70)	77.35 (1.76)	82.53 (0.86)	72.68 (2.86)
	% pCU	37.64 (2.00)	43.03 (5.22)	37.30 (3.89)	44.38 (3.05)	40.00 (2.96)
	AUC	0.9075 (6.66e-03)	0.9274 (1.22e-03)	0.9317 (1.23e-03)	0.9415 (6.57e-04)	0.9070 (6.57e-03)
Criterion		28.79	28.23	27.78	27.76	27.75

Tabla 5.9: Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo GRACE. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva. A=ADAS13, M=MMSE, C=CDRSB, F=FAQ, RF=RAVLT_forgetting.

		RF, M, C, PB	RF, F, M, C, PB	M, C, PB	M, C, L	M, C
Test	Corr. edad	0.9965 (1.72e-03)	0.9966 (3.28e-04)	0.9955 (2.18e-03)	0.9973 (3.58e-04)	0.9962 (9.83e-04)
	Corr. reserva	0.8817 (5.14e-02)	0.8837 (1.26e-02)	0.8516 (6.44e-02)	0.8994 (1.84e-02)	0.8798 (3.03e-02)
	% sCU	85.79 (1.23)	85.42 (1.10)	87.66 (0.39)	90.84 (0.97)	87.01 (0.93)
	% sMCI	89.60 (2.10)	88.12 (1.40)	86.14 (1.32)	94.16 (0.31)	88.91 (1.60)
	% pCU	81.36 (1.44)	78.18 (1.92)	76.82 (1.44)	77.27 (0.00)	76.82 (1.44)
	AUC	0.9373 (1.15e-02)	0.9340 (5.28e-03)	0.9327 (8.79e-03)	0.9406 (5.07e-03)	0.9383 (6.00e-03)
Train	Corr. edad	0.9898 (8.60e-04)	0.9894 (9.25e-04)	0.9914 (5.12e-04)	0.9934 (3.54e-04)	0.9901 (1.07e-03)
	Corr. reserva	0.8291 (1.61e-02)	0.8259 (1.60e-02)	0.8606 (8.93e-03)	0.8526 (7.71e-03)	0.8454 (1.20e-02)
	% sCU	84.62 (0.84)	83.96 (0.78)	84.45 (1.24)	88.28 (0.69)	85.03 (0.67)
	% sMCI	83.62 (2.13)	84.30 (1.69)	82.01 (2.49)	90.05 (0.54)	81.77 (1.82)
	% pCU	72.39 (0.56)	73.26 (1.17)	73.91 (1.62)	68.80 (1.15)	71.09 (1.17)
	AUC	0.9184 (1.33e-02)	0.9138 (1.12e-02)	0.9082 (1.24e-02)	0.9440 (3.81e-03)	0.8837 (1.25e-02)
Criterion		65.62	64.65	64.51	63.85	63.73

Tabla 5.10: Mejores vectores de características para el algoritmo RPDPM. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva. RF=RAVLT_forgetting, M=MMSE, C=CDRSB, PB=mPACctrailsB, F=FAQ, L=LDELTOTAL.

		M, C, PB	M, C, L	M, C	F, C, P	C, P
Test	Corr. edad	0.9955 (2.18e-03)	0.9973 (3.58e-04)	0.9962 (9.83e-04)	0.9952 (1.38e-03)	0.9953 (1.53e-03)
	Corr. reserva	0.8516 (6.44e-02)	0.8994 (1.84e-02)	0.8798 (3.03e-02)	0.8110 (4.85e-02)	0.8208 (5.11e-02)
	% sCU	87.66 (0.39)	90.84 (0.97)	87.01 (0.93)	87.76 (0.69)	88.69 (1.12)
	% sMCI	86.14 (1.32)	94.16 (0.31)	88.91 (1.60)	91.39 (0.67)	91.88 (0.42)
	% pCU	76.82 (1.44)	77.27 (0.00)	76.82 (1.44)	79.09 (3.83)	79.09 (2.35)
	AUC	0.9327 (8.79e-03)	0.9406 (5.07e-03)	0.9383 (6.00e-03)	0.9349 (3.97e-03)	0.9344 (8.39e-03)
Train	Corr. edad	0.9914 (5.12e-04)	0.9934 (3.54e-04)	0.9901 (1.07e-03)	0.9910 (6.42e-04)	0.9912 (4.68e-04)
	Corr. reserva	0.8606 (8.93e-03)	0.8526 (7.71e-03)	0.8454 (1.20e-02)	0.8377 (8.99e-03)	0.8463 (5.52e-03)
	% sCU	84.45 (1.24)	88.28 (0.69)	85.03 (0.67)	86.85 (1.31)	87.86 (0.78)
	% sMCI	82.01 (2.49)	90.05 (0.54)	81.77 (1.82)	86.55 (1.57)	83.98 (1.05)
	% pCU	73.91 (1.62)	68.80 (1.15)	71.09 (1.17)	69.57 (0.51)	67.83 (1.05)
	AUC	0.9082 (1.24e-02)	0.9440 (3.81e-03)	0.8837 (1.25e-02)	0.9183 (1.06e-02)	0.9141 (9.33e-03)
Criterion		64.51	63.85	63.73	61.14	61.05

Tabla 5.11: Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo RPDPM. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto de pCU por correlación con la reserva. M=MMSE, C=CDRSB, PB=mPACCrailsB, L=LDELTOTAL, F=FAQ, P=mPACCDigit.

		RI, RL, C, P, PB	RL, RP, C, P, PB	RL, A, C, P, PB	RF, RP, C, P, PB	RL, C, P, PB, T
Test	Corr. edad	0.9557 (7.96e-04)	0.9557 (7.96e-04)	0.9559 (1.06e-03)	0.9562 (1.22e-03)	0.9557 (7.96e-04)
	Corr. reserva	0.7197 (9.97e-04)	0.7197 (9.97e-04)	0.7193 (1.33e-03)	0.7190 (1.52e-03)	0.7197 (9.97e-04)
	% sCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)
	% sMCI	96.51 (0.74)	95.58 (0.47)	95.81 (0.57)	95.58 (0.47)	95.81 (0.57)
	% pCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)
	AUC	0.9909 (6.37e-04)	0.9914 (6.26e-04)	0.9914 (8.43e-04)	0.9915 (5.76e-04)	0.9911 (4.06e-04)
Train	Corr. edad	0.9713 (4.20e-04)	0.9692 (6.73e-04)	0.9705 (1.16e-03)	0.9703 (3.63e-04)	0.9699 (1.48e-03)
	Corr. reserva	0.8081 (6.31e-03)	0.8041 (5.05e-03)	0.8116 (4.38e-03)	0.8077 (7.68e-03)	0.8059 (8.83e-03)
	% sCU	100.00 (0.00)	99.96 (0.11)	100.00 (0.00)	99.86 (0.24)	99.96 (0.11)
	% sMCI	98.47 (0.17)	98.92 (0.10)	98.54 (0.14)	98.78 (0.23)	98.54 (0.14)
	% pCU	81.76 (1.09)	81.76 (1.09)	80.95 (0.95)	80.95 (1.53)	80.95 (1.76)
	AUC	0.9925 (7.87e-04)	0.9926 (7.50e-04)	0.9928 (8.12e-04)	0.9930 (4.87e-04)	0.9924 (5.11e-04)
Criterion		68.95	68.78	68.75	68.56	68.52

Tabla 5.12: Mejores vectores de características para el algoritmo LSTM. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto de detección de pCU por correlación con la reserva. RI=RAVLT_immediate, RL=RAVLT_learning, RF=RAVLT_forgetting, RP=RAVLT_perc_forgetting, C=CDRSB, A=ADAS13, P=mPACCdigit, PB=mPACCtrailsB, T=TRABSCOR.

		C, P, PB	F, P, PB	F, C, P	RL, F, C	RI, C, P
Test	Corr. edad	0.9559 (1.06e-03)	0.9554 (1.11e-16)	0.9554 (1.11e-16)	0.9556 (8.15e-04)	0.9554 (1.11e-16)
	Corr. reserva	0.7193 (1.33e-03)	0.7200 (0.00e+00)	0.7200 (0.00e+00)	0.7179 (3.47e-03)	0.7200 (0.00e+00)
	% sCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)
	% sMCI	95.81 (0.57)	97.33 (0.53)	96.51 (0.74)	96.16 (0.74)	95.81 (0.77)
	% pCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	98.00 (4.00)	100.00 (0.00)
	AUC	0.9913 (5.42e-04)	0.9849 (1.09e-03)	0.9925 (8.73e-04)	0.9921 (7.34e-04)	0.9910 (6.50e-04)
Train	Corr. edad	0.9683 (1.65e-03)	0.9698 (1.48e-03)	0.9668 (7.00e-04)	0.9667 (1.09e-03)	0.9670 (2.74e-04)
	Corr. reserva	0.7942 (6.85e-03)	0.7962 (5.88e-03)	0.7899 (5.25e-03)	0.7975 (7.15e-03)	0.7903 (1.96e-03)
	% sCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)
	% sMCI	98.02 (0.22)	98.30 (0.10)	98.33 (0.26)	98.37 (0.27)	98.09 (0.23)
	% pCU	77.30 (2.16)	76.89 (0.73)	76.49 (0.90)	77.30 (0.54)	76.08 (0.62)
	AUC	0.9905 (7.09e-04)	0.9860 (4.46e-04)	0.9901 (4.90e-04)	0.9893 (5.12e-04)	0.9900 (2.93e-04)
Criterio		66.45	66.39	65.96	65.86	65.80

Tabla 5.13: Mejores vectores de características de dimensión reducida (2 o 3) para el algoritmo LSTM. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto de detección de pCU por correlación con la reserva. RI=RAVLT_immediate, RL=RAVLT_learning, F=FAQ, C=CDRSB, P=mPACCdigit, PB=mPACCtrailsB.

		RF, M, C, P, PB	RF, C, P, PB	RP, C, P, T	M, C, L, P, PB	M, C, P, PB
Test	RF	1.72 (5.82e-03)	1.72 (5.64e-03)			
	RP			18.04 (2.41e-01)		
	M	1.00 (9.12e-03)			0.95 (8.86e-03)	0.95 (7.13e-03)
	C	0.57 (3.84e-02)	0.62 (4.83e-02)	0.60 (4.09e-02)	0.75 (6.30e-02)	0.64 (7.21e-02)
	L				2.11 (1.13e-02)	
	P	1.61 (5.39e-03)	1.61 (5.52e-03)	1.73 (1.29e-02)	1.58 (5.81e-03)	1.60 (7.15e-03)
	PB	1.42 (5.17e-03)	1.43 (5.56e-03)		1.40 (3.12e-03)	1.40 (7.52e-03)
Train	T			19.62 (7.69e-02)		
	RF	1.66 (4.92e-03)	1.66 (5.48e-03)			
	RP			17.90 (2.01e-01)		
	M	1.01 (8.81e-03)			0.97 (8.88e-03)	0.97 (7.29e-03)
	C	0.82 (7.98e-02)	0.89 (7.96e-02)	0.83 (6.83e-02)	0.99 (1.03e-01)	0.88 (7.19e-02)
	L				2.05 (1.24e-02)	
	P	1.67 (4.62e-03)	1.68 (9.43e-03)	1.85 (1.47e-02)	1.64 (6.71e-03)	1.66 (7.54e-03)
PB	1.44 (6.06e-03)	1.46 (1.03e-02)		1.43 (4.35e-03)	1.43 (1.05e-02)	
	T			20.18 (1.34e-01)		

Tabla 5.14: MAE para los mejores vectores de características entrenados con Leaspy. RF=RAVLT_forgetting, M=MMSE, C=CDRSB, P=mPACCdigit, PB=mPACCtrailsB, RP=RAVLT_perc_forgetting, T=TRABSCOR, L=LDELTOTAL.

		A, C	A, C, PB	A, M, C	A, C, P	RF, A, C
Test	RF					2.01 (1.25e-02)
	A	2.32 (9.36e-03)	2.65 (2.22e-02)	2.33 (1.69e-02)	2.56 (2.11e-02)	2.39 (1.15e-02)
	M			1.16 (1.29e-02)		
	C	0.44 (7.69e-02)	0.46 (6.74e-02)	0.44 (6.35e-02)	0.47 (6.67e-02)	0.43 (6.23e-02)
	P				1.80 (1.83e-02)	
	PB		1.59 (1.71e-02)			
Train	RF					1.95 (1.41e-02)
	A	2.46 (2.45e-02)	2.71 (2.40e-02)	2.43 (2.64e-02)	2.62 (2.35e-02)	2.51 (2.36e-02)
	M			1.17 (9.91e-03)		
	C	0.48 (6.67e-02)	0.66 (9.20e-02)	0.58 (7.46e-02)	0.62 (8.28e-02)	0.51 (7.10e-02)
	P				1.86 (1.62e-02)	
	PB		1.62 (2.22e-02)			

Tabla 5.15: MAE para los mejores vectores de características de dimensión reducida entrenados con Leaspy. A=ADAS13, C=CDRSB, PB=mPACCtrailsB, M=MMSE, RF=RAVLT_forgetting, P=mPACCdigit.

		RF, F, M, C	RF, RP, F, M, C	A, F, M, C	RF, RL, F, M, C	A, F, M, C, P
Test	RF	1.34 (2.68e-03)	1.34 (4.45e-04)		1.35 (8.23e-04)	
	RL				1.13 (4.37e-04)	
	RP		13.52 (5.72e-03)			
	A			1.92 (3.58e-03)		1.92 (1.80e-03)
	F	0.74 (8.78e-03)	0.75 (6.84e-03)	0.74 (9.76e-03)	0.75 (6.04e-03)	0.75 (5.27e-03)
	M	0.85 (7.55e-04)	0.85 (5.59e-04)	0.85 (7.39e-04)	0.85 (8.75e-04)	0.85 (1.36e-03)
	C	0.25 (2.44e-03)	0.25 (3.32e-03)	0.25 (4.48e-04)	0.25 (1.27e-04)	0.25 (2.36e-04)
	P					1.31 (9.64e-04)
Train	RF	1.30 (4.02e-03)	1.29 (7.99e-04)		1.30 (1.87e-03)	
	RL				1.16 (2.86e-04)	
	RP		13.13 (3.49e-03)			
	A			1.94 (8.65e-03)		1.95 (6.40e-03)
	F	0.92 (7.48e-03)	0.93 (5.24e-03)	0.91 (4.81e-03)	0.93 (5.86e-03)	0.92 (3.55e-03)
	M	0.81 (9.61e-04)	0.81 (1.79e-03)	0.81 (2.17e-03)	0.81 (1.86e-03)	0.81 (1.33e-03)
	C	0.27 (9.28e-04)	0.27 (2.74e-03)	0.27 (1.28e-03)	0.27 (9.93e-04)	0.27 (8.39e-04)
	P					1.31 (1.85e-03)

Tabla 5.16: MAE para los mejores vectores de características entrenados con GRACE. RF=RAVLT_forgetting, F=FAQ, M=MMSE, C=CDRSB, RP=RAVLT_perc_forgetting, L=LDELTOTAL, P=mPACCdigit, RL=RAVLT_learning.

		RF, M, C	F, M, C	M, C	F, C, P	RF, A, C
Test	RF	1.35 (6.87e-04)				1.35 (7.02e-04)
	A					1.93 (2.15e-03)
	F	0.73 (6.22e-03)		0.71 (3.07e-03)		
	M	0.85 (6.96e-04)	0.85 (1.22e-03)	0.85 (1.19e-03)		
	C	0.25 (2.55e-03)	0.25 (4.14e-04)	0.25 (1.91e-04)	0.25 (4.95e-04)	0.25 (2.02e-03)
Train	P					1.31 (1.10e-03)
	RF	1.30 (1.99e-03)				1.30 (2.44e-03)
	A					1.97 (2.32e-03)
	F	0.91 (2.40e-03)		0.91 (2.99e-03)		
	M	0.82 (2.33e-03)	0.81 (1.43e-03)	0.81 (2.44e-03)		
	C	0.27 (7.93e-04)	0.27 (5.22e-04)	0.27 (5.26e-04)	0.27 (8.88e-04)	0.27 (2.78e-03)
P					1.32 (3.80e-03)	

Tabla 5.17: MAE para los mejores vectores de características de dimensión reducida entrenados con GRACE. A=ADAS13, M=MMSE, C=CDRSB, F=FAQ, RF=RAVLT_forgetting.

		RF, M, C, PB	RF, F, M, C, PB	M, C, PB	M, C, L	M, C
Test	RF	1.95 (1.91e-02)	1.95 (1.55e-02)			
	F	0.69 (1.56e-02)				
	M	1.13 (5.97e-03)	1.15 (4.41e-03)	1.14 (4.01e-03)	1.16 (1.78e-03)	1.09 (6.28e-03)
	C	0.23 (3.58e-03)	0.28 (3.63e-03)	0.23 (2.92e-03)	0.23 (2.33e-03)	0.22 (1.09e-03)
	L					2.68 (1.47e-02)
Train	T	32.74 (2.78e-01)	32.79 (2.44e-01)	31.06 (2.56e-01)		
	RF	1.92 (1.69e-02)	1.93 (1.74e-02)			
	F	0.68 (1.22e-02)				
	M	1.10 (6.28e-03)	1.11 (5.43e-03)	1.09 (1.03e-02)	1.13 (3.24e-03)	1.03 (7.55e-03)
	C	0.24 (4.62e-03)	0.28 (3.37e-03)	0.24 (6.03e-03)	0.23 (2.07e-03)	0.24 (3.41e-03)
	L					2.61 (1.27e-02)
T	29.92 (3.15e-01)	29.80 (2.85e-01)	28.18 (3.95e-01)			

Tabla 5.18: MAE para los mejores vectores de características entrenados con RPDPM. RF=RAVLT_forgetting, M=MMSE, C=CDRSB, PB=mPACctrailsB, F=FAQ, L=LDELTOTAL.

		M, C, PB	M, C, L	M, C	F, C, P	C, P
Test	F				0.69 (1.19e-02)	
	M	1.14 (4.01e-03)	1.16 (1.78e-03)	1.09 (6.28e-03)		
	C	0.23 (2.92e-03)	0.23 (2.33e-03)	0.22 (1.09e-03)	0.28 (3.50e-03)	0.24 (1.99e-03)
	L		2.68 (1.47e-02)			
	PB				2.04 (1.56e-02)	1.98 (1.22e-02)
Train	T	31.06 (2.56e-01)				
	F				0.66 (7.28e-03)	
	M	1.09 (1.03e-02)	1.13 (3.24e-03)	1.03 (7.55e-03)		
	C	0.24 (6.03e-03)	0.23 (2.07e-03)	0.24 (3.41e-03)	0.28 (2.09e-03)	0.24 (2.12e-03)
	L		2.61 (1.27e-02)			
	PB				1.99 (1.81e-02)	1.95 (1.55e-02)
	T	28.18 (3.95e-01)				

Tabla 5.19: MAE para los mejores vectores de características de dimensión reducida entrenados con RPDPM. M=MMSE, C=CDRSB, PB=mPACCtrailsB, L=LDELTOTAL, F=FAQ, P=mPACCdigit.

		RI, RL, C, P, PB	RL, RP, C, P, PB	RL, A, C, P, PB	RF, RP, C, P, PB	RL, C, P, PB, T
Test	RF				0.32 (7.38e-03)	
	RI	0.17 (4.27e-03)				
	RL	1.34 (1.95e-02)	0.31 (3.08e-03)	0.31 (4.48e-03)		0.32 (2.58e-03)
	RP		1.21 (1.27e-02)		1.19 (2.90e-02)	
	A			1.30 (1.72e-02)		
	C	0.24 (4.14e-03)	0.24 (3.66e-03)	0.24 (3.91e-03)	0.24 (8.66e-03)	1.17 (3.35e-02)
	P	3.55 (9.00e-02)	3.58 (2.55e-01)	3.48 (1.10e-01)	3.59 (1.64e-01)	0.26 (6.06e-03)
	PB	0.70 (2.04e-02)	0.68 (3.33e-02)	0.67 (1.31e-02)	0.69 (2.98e-02)	4.06 (9.73e-02)
	T					0.62 (1.45e-02)
Train	RF				0.26 (7.10e-03)	
	RI	0.19 (1.23e-03)				
	RL	1.40 (1.10e-02)	0.33 (3.91e-03)	0.33 (2.99e-03)		0.34 (4.31e-03)
	RP		1.09 (2.00e-02)		1.10 (3.22e-02)	
	A			0.93 (1.15e-02)		
	C	0.17 (3.75e-03)	0.17 (3.69e-03)	0.17 (2.99e-03)	0.18 (9.43e-03)	0.86 (3.76e-02)
	P	2.50 (6.83e-02)	2.56 (2.58e-01)	2.49 (1.43e-01)	2.57 (2.10e-01)	0.20 (6.29e-03)
	PB	0.52 (2.76e-02)	0.50 (4.07e-02)	0.49 (1.98e-02)	0.52 (4.39e-02)	3.13 (9.70e-02)
	T					0.47 (1.10e-02)

Tabla 5.20: MAE para los mejores vectores de características entrenados con LSTM. RI=RAVLT_immediate, RL=RAVLT_learning, RF=RAVLT_forgetting, RP=RAVLT_perc_forgetting, C=CDRSB, A=ADAS13, P=mPACCdigit, PB=mPACCtrailsB, T=TRABSCOR.

		C, P, PB	F, P, PB	F, C, P	RL, F, C	RI, C, P
Test	RI					0.17 (5.31e-03)
	RL				0.32 (5.40e-03)	
	F		0.24 (8.78e-03)	0.27 (5.98e-03)	1.14 (3.51e-02)	
	C	0.28 (2.35e-02)		1.27 (4.14e-02)	0.26 (6.74e-03)	1.21 (2.89e-02)
	P	1.33 (1.74e-01)	1.08 (1.05e-01)	0.24 (6.18e-03)		0.24 (4.29e-03)
	PB	0.31 (2.54e-02)	0.26 (2.17e-02)			
Train	RI					0.20 (5.11e-03)
	RL				0.34 (5.50e-03)	
	F		0.21 (1.02e-02)	0.23 (8.22e-03)	0.96 (4.59e-02)	
	C	0.21 (2.22e-02)		0.99 (4.47e-02)	0.20 (4.55e-03)	0.88 (2.96e-02)
	P	1.08 (1.98e-01)	0.84 (1.11e-01)	0.18 (5.57e-03)		0.18 (3.91e-03)
	PB	0.26 (3.68e-02)	0.21 (3.20e-02)			

Tabla 5.21: MAE para los mejores vectores de características de dimensión reducida entrenados con LSTM. RI=RAVLT_immediate, RL=RAVLT_learning, F=FAQ, C=CDRSB, P=mPACCdigit, PB=mPACCtrailsB.

5.4. DPS y tiempos de conversión

Según se ha comentado anteriormente, el vector que mejor modela la fase preclínica según Leaspy teniendo en cuenta su dimensión y composición es:

$$\{ ADAS13, CDRSB \}$$

Según GRACE y RPDPM, se considera:

$$\{ MMSE, CDRSB \}$$

Serán, por lo tanto, los vectores para los cuales se realizarán los análisis posteriores. De esta forma, se establece un marco de referencia para comparar modelos de progresión de la enfermedad.

Los resultados para los vectores AC (Leaspy) y MC (GRACE) se exponen a continuación, donde se han incluido los datos de *train* y *test*. La figura 5.5 representa el número de sujetos frente al DPS estimado por GRACE, RPDPM y Leaspy. La población se ha clasificado según su diagnóstico al inicio del estudio. Si la población se clasifica, además, en estable o progresiva los histogramas de la figura 5.6 representa la distribución. La enfermedad está más avanzada cuanto mayor sea el valor del DPS. Los sujetos CU y SMC se encuentran desplazados hacia la izquierda del eje de abscisas y los sMCI a la derecha. Esto es coherente con lo esperado, pues al inicio del estudio los CU y SMC están cognitivamente normales, mientras que los sMCI ya han pasado la barrera del deterioro cognitivo. El error en la clasificación es significativamente mayor en GRACE, ya que hay más solapamientos entre grupos.

Cabe destacar que, en todos los casos, los sujetos progresivos están más cerca del deterioro cognitivo que los estables, especialmente en el caso de los pSMC. Además, los sujetos con queja subjetiva (tanto estables como progresivos) se encuentran muy ligeramente desplazados hacia la derecha con respecto a los cognitivamente normales.

Como era de esperar, tanto Leaspy como funcionan mejor que GRACE. Las distribuciones de sujetos devuelta por Leaspy están muy concentrada en torno a $DPS = 0$. Lo mismo sucede con RPDPM alrededor de 0.5. Sin embargo, ambos discriminan mejor entre sujetos cognitivamente normales y con deterioro cognitivo. Consecuentemente, el clasificador de Bayes tendrá una elevada capacidad de clasificación, y las medidas de correlación con la reserva y detección de pCU serán altas.

La figura 5.7 muestra la reserva cognitiva estimada por GRACE, RPDPM y Leaspy frente a la real. Se han usado los sujetos progresivos (pCU y pSMC), ya que sus tiempos de conversión reales son conocidos (no se sabe cuándo convertirán los sCU y sSMC, ni cuándo convirtieron los sMCI). El clasificador de Bayes estima el grupo al que pertenece cada visita. El tiempo de conversión corresponde al valor medio de las edades entre las que el valor de su etiqueta cambia de cognitivamente normal a deterioro cognitivo. Cada punto corresponde a la media de las estimaciones de 10 modelos (*bootstrap*=10). Adicionalmente, se ha trazado la recta de regresión que mejor estima la nube de puntos. Una recta ideal sería aquella del tipo $y = x + 0$ con un coeficiente de Pearson igual a la unidad. La pendiente unitaria indica que el error no aumenta cuanto más lejana sea la estimación. El intercepto nulo es indicador de la desviación inicial. Este último no es relevante porque se puede corregir manualmente sin afectar a los resultados: solamente desplaza la recta

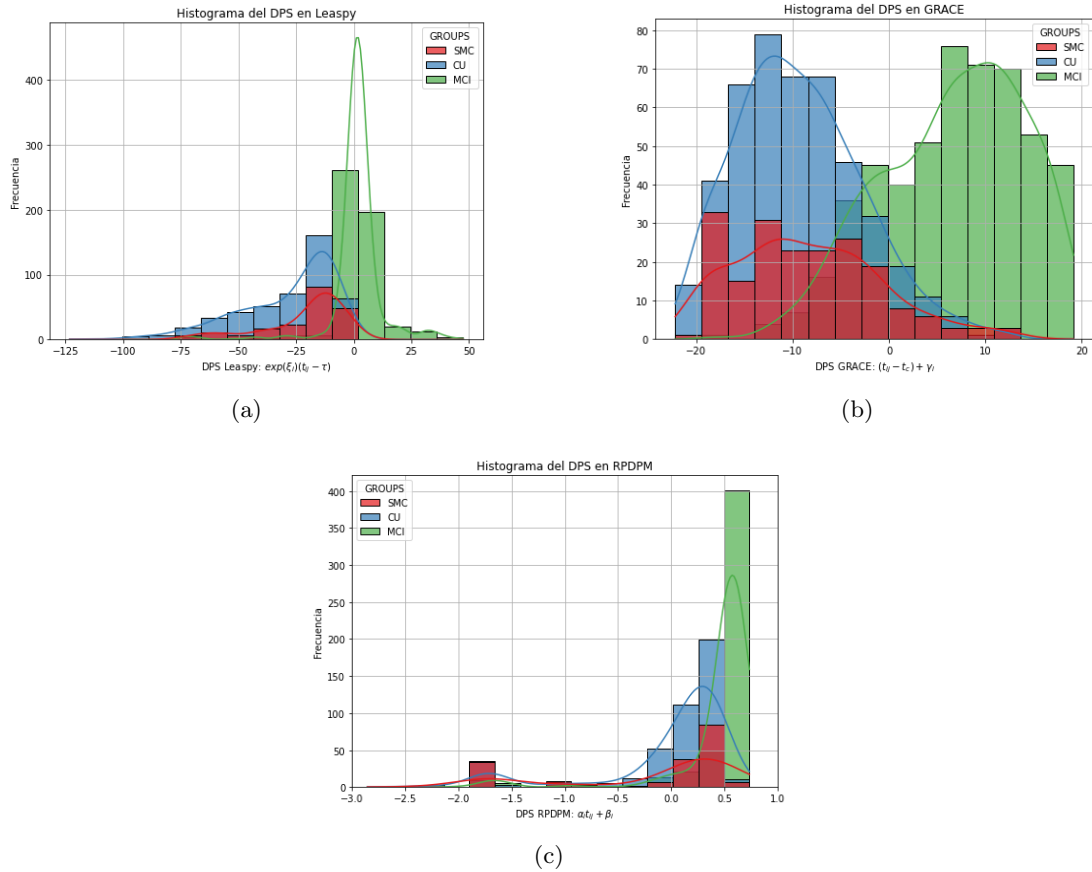


Figura 5.5: Histogramas con los tiempos de conversión estimados por Leaspy para ADAS13, CDRSB (a); GRACE (b) y RPDPM (c) para MMSE, CDRSB. La población se agrupa en CU, SMC y sMCI.

junto con la nube de puntos.

Las rectas obtenidas son $y = 0,87x - 0,20$ en Leaspy, $y = 0,97x - 0,62$ en GRACE, e $y = 0,87x - 0,19$ en RPDPM. Los coeficientes de correlación son $R = 0.8012$, $R = 0.6708$ y $R = 0.7732$, respectivamente. En todos los casos se obtienen rectas con una pendiente muy próxima a la unitaria. En Leaspy y RPDPM, el intercepto es cercano al ideal. Las estimaciones realizadas por Leaspy son las mejores, ya que se obtiene un coeficiente de correlación mayor.

La figura 5.8 representa las estimaciones de reserva cognitiva entre los algoritmos propuestos. Se han representado los grupos pCU y pSMC. Las rectas de regresión tienen las formas $y = 0,79x - 0,07$ (GRACE - Leaspy), $y = 0,80x - 0,24$ (GRACE - RPDPM) e $y = 0,88x - 0,22$ (RPDPM - Leaspy); con $R = 0.7382$, $R = 0.7317$ y $R = 0.8083$, respectivamente. Como en el caso anterior, idealmente la recta sería $y = x + 0$ con $R = 1$, pues implicaría que por ambas vías se llega al mismo resultado. Las pendientes son cercana a la unidad, lo que indica que la diferencia entre los algoritmos no es muy grande cuando las estimaciones son lejanas en el tiempo. Los interceptos son próximos a 0, especialmente en el caso GRACE - Leaspy, por lo que no hay una desviación fija entre estimaciones. Los coeficientes de regresión son altos, implicando una alta correlación.

Es importante resaltar que los resultados arrojados por un vector dado son diferentes entre algoritmos. Una determinada combinación puede funcionar correctamente en

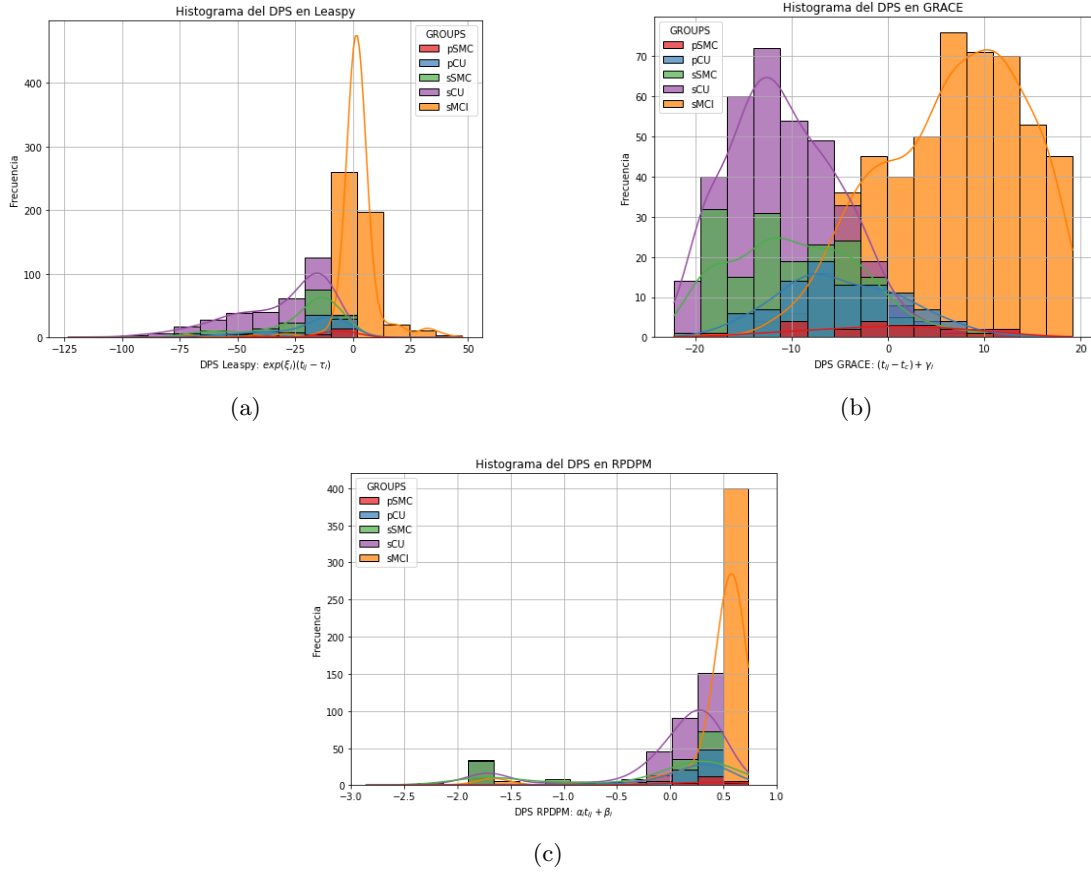


Figura 5.6: Histogramas con los tiempos de conversión estimados por Leaspy para ADAS13, CDRSB (a); GRACE (b) y RPDPM (c) para MMSE y CDRSB . La población se agrupa en sCU, pCU, sSMC, pSMC, sMCI.

Leaspy, pero no en GRACE o RPDPM, y viceversa. Cada algoritmo debería usarse con la combinación que maximice sus resultados, con objeto de obtener las mejores predicciones posibles.

5.5. Robustez frente a valores faltantes

Los vectores de características AC (Leaspy), MC (GRACE y RPDPM) y R_LFC (LSTM) se evalúan de nuevo introduciendo NaN de manera intencionada. De esta forma, se simula la falta de marcadores en las visitas de los pacientes.

Las tablas 5.22, 5.23, 5.24 y 5.25 muestran los resultados de los vectores AC, MC, MC y R_LFC entrenados con Leaspy, GRACE, RPDPM y LSTM, respectivamente, para los porcentajes de NaN iniciales (1.25 % en AC, 0.90 % en MC-GRACE, 0.88 % en MC-RPDPM³ y 1.21 % en R_LFC -LSTM), 10 %, 20 % y 40 %. Los errores medios absolutos se presentan en las tablas 5.26, 5.27, 5.28 y 5.29 para los mismos porcentajes.

Los resultados empeoran con todos los algoritmos al aumentar la cantidad de NaN introducidos. En Leaspy, la detección de pCU y la correlación con la reserva disminuyen

³Los porcentajes iniciales de NaN difieren entre RPDPM y GRACE a pesar de emplearse las mismas características. Esto es porque en RPDPM se ha usado una semilla aleatoria al dividir los datos en *train* y *test*. Por la misma razón los resultados de RPDPM son ligeramente distintos a los presentados en la tabla del análisis masivo.

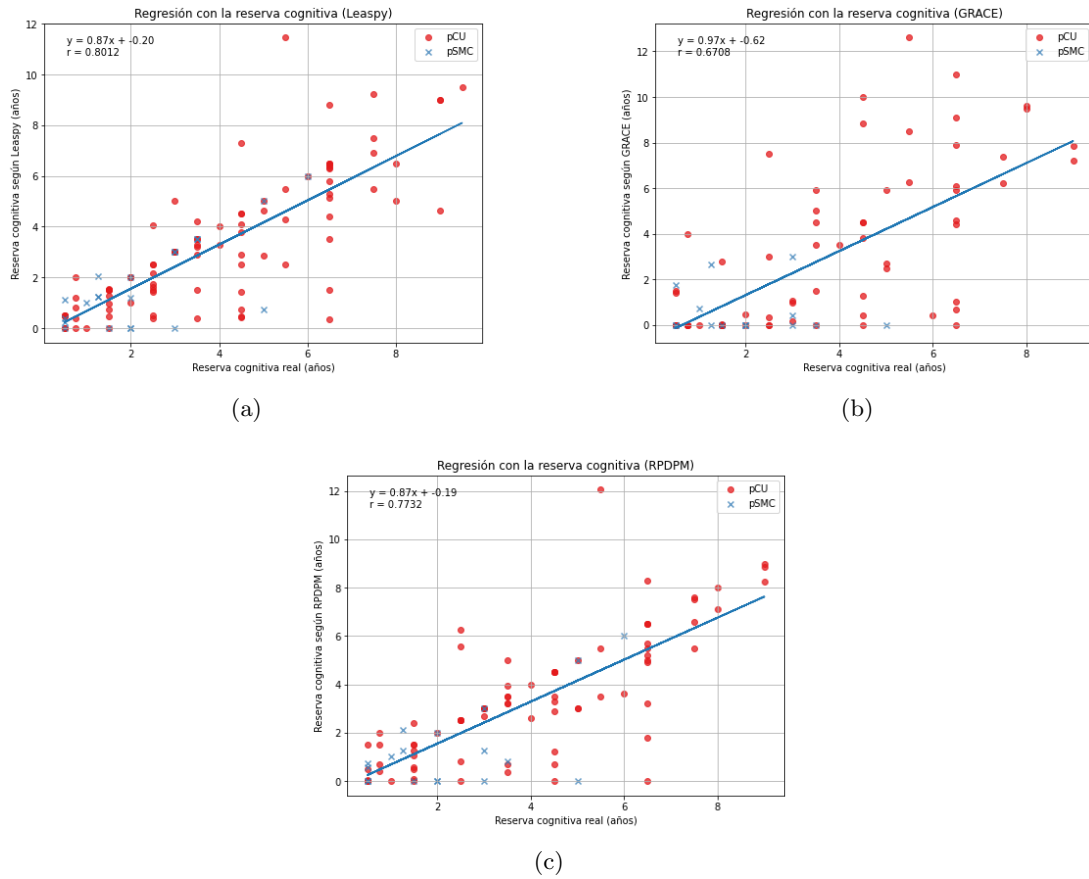


Figura 5.7: Gráfico de dispersión con tiempo de conversión real frente a tiempos estimados por Leaspy para ADAS13, CDRSB (a); GRACE (b) y RPDPM (c) para MMSE, CDRSB. Sujetos pCU y pSMC.

notablemente: de 80.77 % a 68.08 % y de 0.7806 a 0.7187 (de 83.07 % a 67.05 % y de 0.8576 a 0.7966 en *train*), respectivamente. Ocurre lo mismo en GRACE: de 57.60 % a 30.80 % y de 0.5634 a 0.4487 (de 37.30 % a 25.39 % y de 0.6375 a 0.4566 en *train*). La detección de pCU se ve significativamente afectada en RPDPM, mientras que la correlación con las reserva es más estable: de 76.82 % a 55.91 % y de 0.8730 a 0.8675⁴ (de 71.09 % a 61.96 % y de 0.8457 a 0.8359 en *train*). Ambas métricas se mantienen estables en *train* al usar LSTM, mientras que en *test* experimentan una mayor disminución: de 96.00 % a 82.00 % y de 0.7162 a 0.6159⁵. Los valores del criterio en Leaspy caen ligeramente entre el 1.25 % de NaN y el 20 % de NaN, y sufren un descenso apreciable a partir del 40 % de NaN. RPDPM y LSTM siguen la misma tendencia. En GRACE, este descenso ocurre en 2 etapas: a partir del 10 % de NaN y del 40 % de NaN.

En cuanto a los MAE, se mantienen constantes en Leaspy. En RPDPM se observa un descenso progresivo de su valor en MMSE, mientras que CDRSB apenas varía. Una mejora similar se aprecia en GRACE tanto en MMSE como en CDRSB. Lo mismo ocurre en LSTM para todos los marcadores.

Se concluye que los algoritmos en orden decreciente de robustez son: LSTM, RPDPM,

⁴Nótese que la correlación con la reserva baja a 0.7700 en *test* para un 20 % de NaN. La explicación podría encontrarse en el *split* de los datos, ya que en *train* este valor es 0.8252 para el mismo porcentaje de NaN.

⁵Los valores para el porcentaje inicial de NaN en LSTM no son exactamente iguales a los del análisis masivo debido a que entre ejecuciones pueden existir diferencias en el entrenamiento.

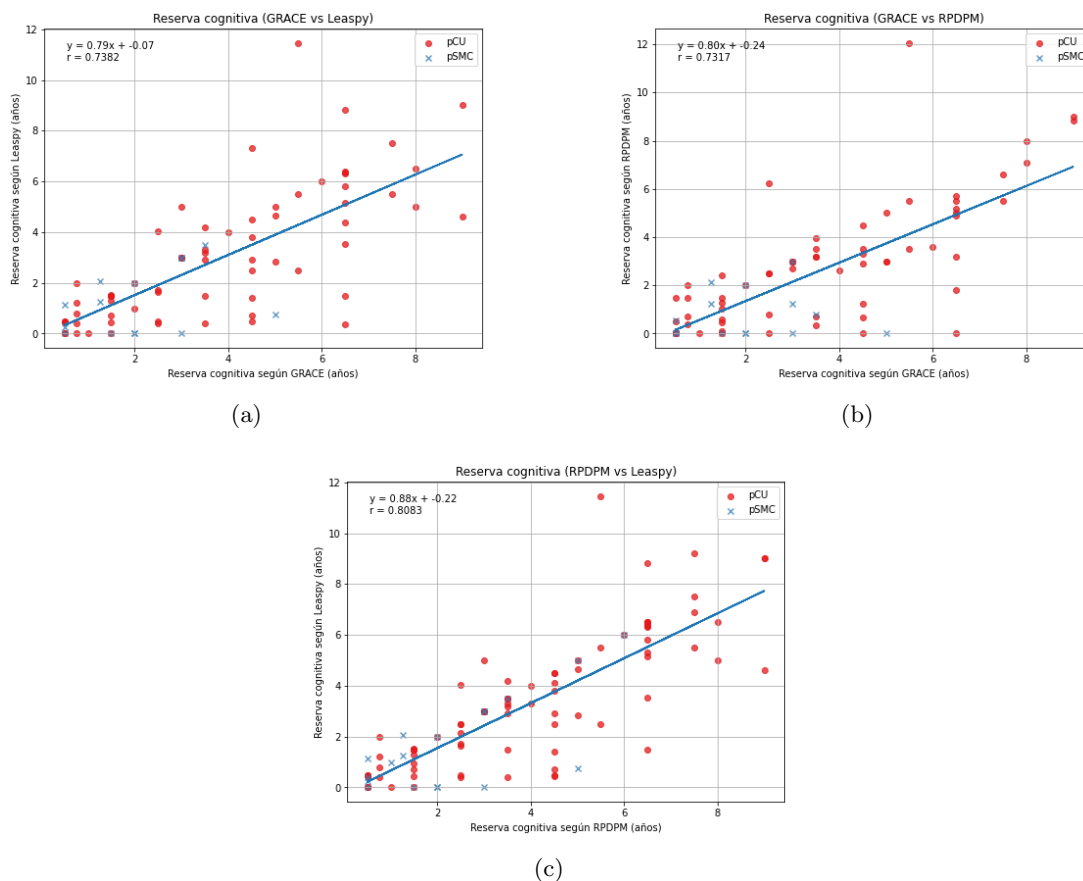


Figura 5.8: Gráfico de dispersión con las estimaciones de tiempos de conversión entre los algoritmos propuestos: GRACE - Leaspy (a), GRACE - RPDPM (c) y RPDPM - Leaspy (c). Leaspy se ha entrenado con ADAS13, CDRSB; GRACE y RPDPM con MMSE, CDRSB. Sujetos pCU y pSMC.

Leaspy, GRACE⁶.

⁶Entre Leaspy y RPDPM no está clara la diferencia en robustez. Atendiendo únicamente a los valores de las métricas y el criterio en los distintos porcentajes de NaN, Leaspy supera a RPDPM. No obstante, en términos relativos RPDPM es más robusto. No se debe olvidar que Leaspy presenta un mayor desempeño que RPDPM, por lo que este se ve beneficiado de una comparativa absoluta.

		1.25 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	Corr. edad	0.9887 (1.58e-03)	0.9876 (6.67e-04)	0.9875 (1.07e-03)	0.9854 (1.02e-03)
	Corr. reserva	0.7806 (2.69e-02)	0.7612 (1.32e-02)	0.7672 (2.33e-02)	0.7187 (3.02e-02)
	% sCU	75.14 (1.40)	76.26 (1.52)	75.05 (4.18)	76.17 (1.97)
	% sMCI	94.30 (1.10)	92.70 (0.90)	94.30 (0.64)	92.80 (1.47)
	% pCU	80.77 (5.44)	79.62 (3.46)	75.38 (8.63)	68.08 (7.70)
	AUC	0.9487 (1.25e-02)	0.9453 (1.16e-02)	0.9400 (9.74e-03)	0.9301 (1.06e-02)
Train	Corr. edad	0.9924 (1.21e-03)	0.9930 (9.17e-04)	0.9915 (1.93e-03)	0.9905 (1.57e-03)
	Corr. reserva	0.8576 (2.05e-02)	0.8587 (1.75e-02)	0.8316 (4.62e-02)	0.7966 (3.56e-02)
	% sCU	79.16 (1.72)	79.70 (1.21)	75.20 (3.78)	76.27 (2.37)
	% sMCI	95.16 (1.15)	94.00 (1.05)	95.35 (0.28)	92.42 (0.69)
	% pCU	83.07 (7.18)	80.11 (3.57)	76.25 (9.82)	67.05 (6.22)
	AUC	0.9665 (5.42e-03)	0.9656 (3.39e-03)	0.9596 (6.00e-03)	0.9425 (7.18e-03)
Criterion		67.02	64.57	60.55	51.12

Tabla 5.22: Medidas de calidad para el vector ADAS13, CDRSB entrenado con Leaspy al introducir valores NaN. El *set* de datos original incluye un 1.25 % de NaN. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva.

		0.90 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	Corr. edad	0.9801 (1.27e-03)	0.9757 (1.58e-03)	0.9779 (1.67e-03)	0.9650 (3.32e-03)
	Corr. reserva	0.5634 (4.18e-02)	0.4788 (2.37e-02)	0.5051 (4.58e-02)	0.4487 (5.07e-02)
	% sCU	77.71 (2.77)	77.62 (2.67)	77.62 (1.63)	58.76 (1.85)
	% sMCI	75.69 (5.69)	71.67 (2.67)	72.25 (2.02)	55.59 (1.04)
	% pCU	57.60 (6.31)	36.00 (4.22)	39.20 (5.59)	30.80 (6.55)
	AUC	0.9225 (1.50e-03)	0.9093 (1.74e-03)	0.9031 (7.53e-04)	0.7480 (1.12e-03)
Train	Corr. edad	0.9805 (7.19e-04)	0.9804 (1.01e-03)	0.9641 (1.54e-03)	0.9677 (1.34e-03)
	Corr. reserva	0.6375 (2.22e-02)	0.6355 (1.90e-02)	0.4813 (1.61e-02)	0.4566 (5.06e-02)
	% sCU	82.06 (2.29)	80.60 (1.54)	61.81 (1.23)	57.19 (1.14)
	% sMCI	77.35 (1.76)	75.18 (1.30)	54.79 (0.71)	53.53 (1.21)
	% pCU	37.30 (3.89)	36.07 (1.71)	31.12 (1.68)	25.39 (3.19)
	AUC	0.9317 (1.23e-03)	0.9166 (1.55e-03)	0.6823 (1.32e-03)	0.6540 (1.64e-03)
Criterion		27.78	19.88	17.22	12.66

Tabla 5.23: Medidas de calidad para el vector MMSE, CDRSB entrenado con GRACE al introducir valores NaN. El *set* de datos original incluye un 0.90 % de NaN. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva.

		0.88 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	Corr. edad	0.9960 (9.12e-04)	0.9947 (6.45e-04)	0.9895 (1.78e-03)	0.9951 (1.75e-03)
	Corr. reserva	0.8730 (2.72e-02)	0.8271 (1.82e-02)	0.7700 (3.11e-02)	0.8675 (3.27e-02)
	% sCU	87.48 (1.00)	84.67 (0.90)	82.80 (1.18)	78.22 (1.33)
	% sMCI	88.91 (1.80)	87.82 (1.05)	85.05 (1.19)	77.23 (2.91)
	% pCU	76.82 (1.44)	76.36 (1.92)	72.73 (0.00)	55.91 (2.20)
	AUC	0.9409 (6.70e-03)	0.9291 (6.17e-03)	0.9156 (3.38e-03)	0.8950 (6.47e-03)
Train	Corr. edad	0.9901 (6.88e-04)	0.9898 (1.19e-03)	0.9888 (8.86e-04)	0.9917 (1.59e-03)
	Corr. reserva	0.8457 (9.49e-03)	0.8294 (1.52e-02)	0.8252 (1.39e-02)	0.8359 (3.53e-02)
	% sCU	85.08 (0.49)	84.27 (0.51)	84.36 (0.67)	77.65 (0.82)
	% sMCI	81.75 (1.34)	81.99 (1.61)	77.26 (1.34)	74.03 (2.49)
	% pCU	71.09 (0.56)	67.39 (0.51)	71.63 (0.80)	61.96 (1.54)
	AUC	0.8822 (1.01e-02)	0.8977 (1.22e-02)	0.8755 (1.36e-02)	0.8718 (1.51e-02)
Criterion		63.50	59.42	57.53	50.12

Tabla 5.24: Medidas de calidad para el vector MMSE, CDRSB entrenado con RPDPM al introducir valores NaN. El *set* de datos original incluye un 0.88 % de NaN. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva.

		1.21 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	Corr. edad	0.9556 (8.31e-04)	0.9556 (8.38e-04)	0.9555 (8.51e-04)	0.9358 (7.50e-03)
	Corr. reserva	0.7162 (4.12e-03)	0.7153 (4.14e-03)	0.7136 (3.59e-03)	0.6159 (3.36e-02)
	% sCU	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)
	% sMCI	96.05 (0.93)	96.74 (0.47)	96.40 (0.35)	97.67 (0.00)
	% pCU	96.00 (4.90)	95.00 (5.00)	93.00 (4.58)	82.00 (6.00)
	AUC	0.9922 (6.43e-04)	0.9939 (3.72e-04)	0.9887 (1.18e-03)	0.9876 (1.34e-03)
Train	Corr. edad	0.9666 (1.38e-03)	0.9666 (1.69e-03)	0.9653 (1.80e-03)	0.9624 (3.30e-03)
	Corr. reserva	0.7950 (9.61e-03)	0.7955 (9.71e-03)	0.7904 (8.14e-03)	0.7718 (2.02e-02)
	% sCU	100.00 (0.00)	100.00 (0.00)	99.96 (0.11)	99.71 (0.27)
	% sMCI	98.33 (0.30)	98.23 (0.33)	98.54 (0.14)	98.65 (0.10)
	% pCU	77.30 (0.54)	76.89 (0.95)	75.14 (1.08)	76.62 (1.71)
	AUC	0.9895 (4.77e-04)	0.9880 (4.16e-04)	0.9858 (1.23e-03)	0.9830 (9.81e-04)
Criterion		65.00	64.47	62.78	54.65

Tabla 5.25: Medidas de calidad para el vector RAVLT_learning, FAQ, CDRSB entrenado con LSTM al introducir valores NaN. El *set* de datos original incluye un 1.21 % de NaN. Corr. edad=Correlación con la edad, Corr. reserva=Correlación con la reserva, % sCU=Detección de sCU, % pCU=Detección de pCU, % sMCI=Detección de sMCI, AUC=Área bajo la curva. El criterio es la media geométrica entre *train* y *test* del producto detección de pCU por correlación con la reserva.

		1.25 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	A	2.32 (9.78e-03)	2.32 (1.08e-02)	2.33 (3.24e-02)	2.29 (2.04e-02)
	C	0.47 (8.18e-02)	0.42 (5.63e-02)	0.50 (4.02e-02)	0.49 (6.71e-02)
Train	A	2.45 (2.98e-02)	2.44 (2.31e-02)	2.40 (3.63e-02)	2.30 (3.47e-02)
	C	0.53 (5.69e-02)	0.47 (5.77e-02)	0.57 (6.62e-02)	0.57 (1.15e-01)

Tabla 5.26: MAE para el vector ADAS13, CDRSB entrenado con Leaspy al introducir valores NaN. El *set* de datos original incluye un 1.25 % de NaN. A=ADAS13, C=CDRSB.

		0.90 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	M	0.85 (1.19e-03)	0.88 (1.29e-03)	0.79 (2.06e-04)	0.64 (2.74e-03)
	C	0.25 (1.91e-04)	0.25 (3.47e-04)	0.23 (4.47e-04)	0.19 (3.05e-03)
Train	M	0.81 (2.44e-03)	0.79 (2.40e-03)	0.75 (3.88e-03)	0.60 (1.86e-03)
	C	0.27 (5.26e-04)	0.27 (7.03e-04)	0.25 (4.68e-04)	0.21 (9.59e-04)

Tabla 5.27: MAE para el vector MMSE, CDRSB entrenado con GRACE al introducir valores NaN. El *set* de datos original incluye un 0.90 % de NaN. M=MMSE, C=CDRSB.

		0.88 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	M	1.09 (6.19e-03)	1.08 (7.57e-03)	1.06 (4.27e-03)	1.00 (1.25e-02)
	C	0.22 (2.28e-03)	0.22 (3.61e-03)	0.24 (2.54e-03)	0.23 (8.02e-03)
Train	M	1.03 (7.15e-03)	1.03 (1.04e-02)	0.99 (1.45e-02)	0.94 (1.26e-02)
	C	0.24 (3.42e-03)	0.24 (4.27e-03)	0.24 (6.96e-03)	0.25 (9.39e-03)

Tabla 5.28: MAE para el vector MMSE, CDRSB entrenado con RPDPM al introducir valores NaN. El *set* de datos original incluye un 0.88 % de NaN. M=MMSE, C=CDRSB.

		1.21 % de NaN	10 % de NaN	20 % de NaN	40 % de NaN
Test	RL	0.28 (4.12e-03)	0.26 (5.39e-03)	0.26 (4.70e-03)	0.24 (7.52e-03)
	F	0.27 (1.04e-02)	0.24 (7.16e-03)	0.19 (7.63e-03)	0.23 (9.58e-03)
	C	0.10 (5.33e-03)	0.10 (4.26e-03)	0.11 (3.96e-03)	0.09 (2.77e-03)
Train	RL	0.31 (4.42e-03)	0.28 (6.93e-03)	0.27 (4.06e-03)	0.19 (6.29e-03)
	F	0.23 (1.02e-02)	0.22 (9.69e-03)	0.18 (1.08e-02)	0.18 (1.07e-02)
	C	0.08 (5.24e-03)	0.08 (4.23e-03)	0.07 (3.54e-03)	0.05 (3.31e-03)

Tabla 5.29: MAE para el vector RAVLT_learning, FAQ, CDRSB entrenado con LSTM al introducir valores NaN. El *set* de datos original incluye un 0.88 % de NaN. RL=RAVLT_learning, F=FAQ, C=CDRSB.

5.6. Análisis frente al número de visitas

Se evalúan las medidas de calidad de GRACE (MC), RPDPM (MC) y Leaspy (AC) en función del número de visitas de los sujetos. El análisis se ha realizado sobre la combinación de los conjuntos de *train* y *test*. Los valores se han calculado considerando *bootstrap* = 10 para aumentar la estabilidad de las estimaciones.

El área bajo la curva, los porcentajes de sCU, sMCI y pCU, y las correlaciones con la edad y la reserva se presentan en los diagramas de barras de la figura 5.9. No es sorprendente que Leaspy y RPDPM superen a GRACE en casi todas las puntuaciones, independientemente del número de visitas. En todos los algoritmos⁷, el AUC, %sCU y %sMCI descienden progresivamente al aumentar el número de visitas. El %pCU, por el contrario, sigue la tendencia opuesta. La razón de esto, es la parametrización de los modelos. Al imponerse curvas monotonamente crecientes, es más probable que el diagnóstico sea CU en las visitas más cercanas a la inicial y MCI en las últimas visitas. Debido a que el periodo seguimiento de los sujetos con más visitas es mayor, serán clasificados como conversores con mayor probabilidad. Opuestamente, los sujetos con pocas visitas serán mayormente clasificados como estables. Siguiendo con el mismo razonamiento, será más difícil clasificar a un sujeto como sMCI cuantas más visitas tenga. Por lo tanto, las métricas más reveladoras son el %sCU y %sMCI en sujetos con más visitas y el %pCU en sujetos con menos visitas. En este sentido, Leaspy y RPDPM presentan unas puntuaciones muy superiores a las de GRACE. Aunque la diferencia no sea evidente entre los dos primeros, Leaspy se impone en la detección de pCU y sMCI, manteniéndose competitivo en los %sCU.

Por otro lado, resulta interesante analizar conjuntamente la detección de pCU y la correlación con la reserva⁸. Aunque la tendencia sea clasificar como progresivos a los sujetos con muchas visitas, una elevada correlación con la reserva será indicador de que las estimaciones del modelo son precisas. Por ejemplo, Leaspy clasifica correctamente a casi todos los sujetos con 9 o 10 visitas, y además es capaz de determinar el momento en que lo hace con bastante precisión (el coeficiente de Pearson con la reserva es cercano a 0.9). No así, GRACE detecta mejor (aunque peor que Leaspy) a los sujetos progresivos con muchas que los que tienen pocas visitas, pero su coeficiente de correlación con la reserva no supera 0.6.

Los errores medios absolutos se presentan en los diagramas de caja y bigotes de la figura 5.10, donde se han omitido los *outliers*. Debido a que los modelos utilizan distintos marcadores, solamente es posible comparar aquellos que sean comunes. Para los demás, el análisis es únicamente frente al número de visitas y no respecto a otros modelos.

El único marcador en común entre Leaspy, GRACE y RPDPM es CDRSB. Como era de esperar, para todas las visitas los errores de GRACE y RPDPM son menores. Para CDRSB y el resto de marcadores, se observa que las medianas y el cuartil inferior se mantienen más o menos constantes. El cuartil superior aumenta con el número de visitas, especialmente en el caso de CDRSB. Los bigotes superiores se alargan siguiendo la misma tendencia. RPDPM sigue estos patrones con menor claridad. Se concluye que la dependencia del ajuste espacial es débil con el número de visitas, pero a medida que estas aumentan, es más propensa la aparición de valores atípicos. Esto implica que el error espacial en el ajuste de la curva se vuelve menor al descender las visitas. Especialmente, el error con 2 visitas es nulo en GRACE debido a que la personalización consiste en trazar

⁷Nótese que la tendencia parece romperse en las últimas barras, debido a la escasez de sujetos con tales cantidades de visitas.

⁸Nótese que el %pCU y la correlación con la reserva son las métricas con mayor variabilidad.

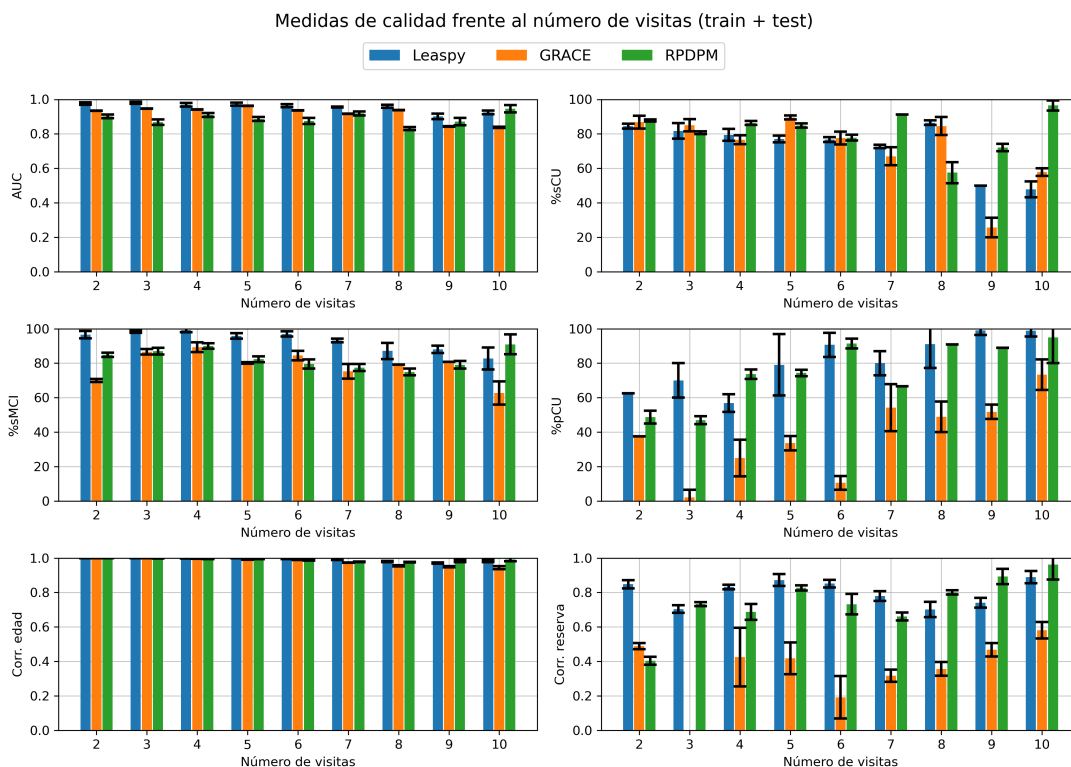


Figura 5.9: Diagramas de barras de las medidas de calidad frente al número de visitas. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y MMSE, CDRSB (MC) para GRACE y RPDPM. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

la recta de mínimos cuadrados.

5.7. Pronóstico

Los vectores de características AC (Leaspy), MC (GRACE), MC (RPDPM) y R_LFC (LSTM) se utilizan para estimar la evolución de los sujetos en el futuro. Las últimas visitas de los pacientes se eliminan sucesivamente, realizando la personalización con el resto. En los métodos paramétricos, se impone como condición de diseño que la mitad del total de las visitas de cada sujeto debe ser mayor al número de visitas a predecir. En las redes neuronales recurrentes, el número de visitas debe ser mayor o igual a la suma de la longitud de secuencia, más dos visitas que determinan el diagnóstico durante la personalización, más tantas visitas como puntos futuros se desee predecir.

Las medidas de calidad del pronóstico referentes a %sCU, %pCU, %sMCI, AUC, BCA se presentan en los diagramas de barras de las figuras 5.11 y 5.12. Estas comparan los métodos paramétricos entre sí y las redes neuronales recurrentes con Leaspy⁹, respectivamente. Las correlaciones con la edad y la reserva, junto con el %pCU, se presentan en las figuras 5.13 y 5.14. En ambos casos, se ha utilizado el conjunto de la población de *train* con la de *test*. Conviene destacar que el %pCU representado en ambas gráficas no se trata de la misma medida (de ahí que sus valores no coincidan). Esto es debido al método seguido para realizar ambas representaciones:

- Figuras 5.11 y 5.12: el modelo predice las últimas visitas de los pacientes, usando el

⁹Se escoge Leaspy por ser el mejor entre los métodos paramétricos. No es necesario comparar las RNN con el resto de algoritmos.

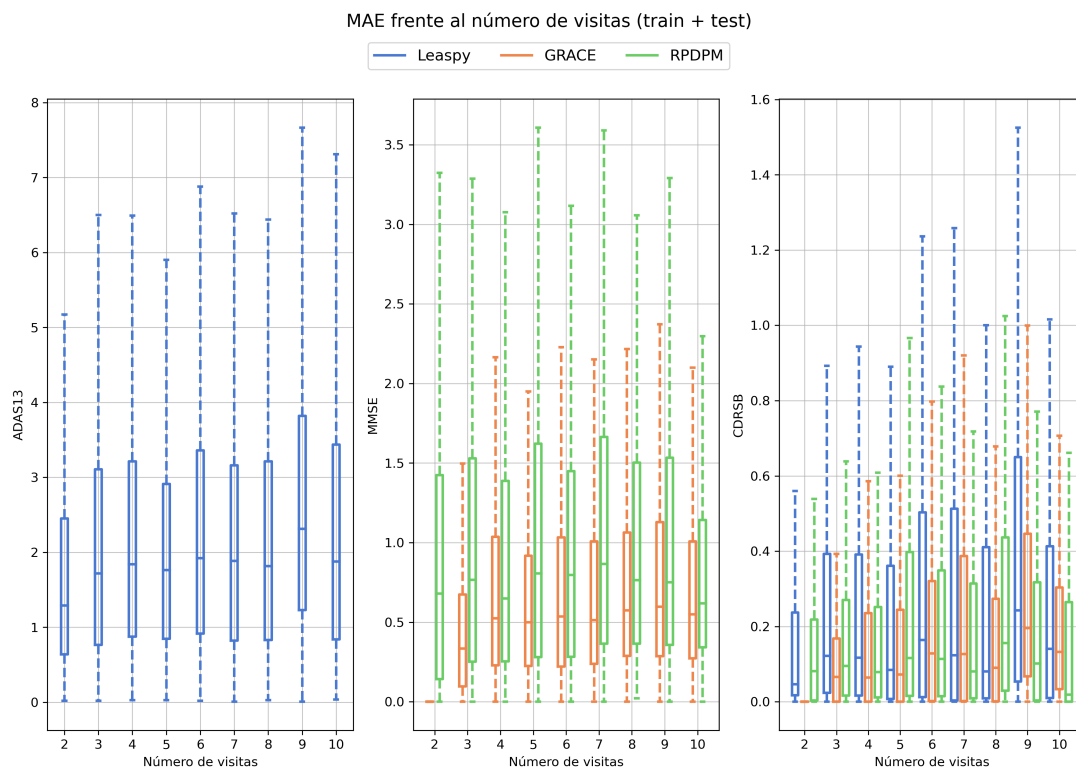


Figura 5.10: Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente al número de visitas. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y MMSE, CDRSB (MC) para GRACE y RPDPM. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

resto para la personalización. La clase pCU se define como los sujetos CU estables durante la personalización que convierten a MCI en el futuro. El %pCU refleja la relación entre el número de sujetos pCU estimados por el modelo y el real.

- Figuras 5.13 y 5.14: tiene en cuenta solamente los sujetos pCU reales (según la definición para diagnóstico). Los modelos estiman el diagnóstico entre la primera visita¹⁰ y 10 años después de la última usada en la personalización. Se personaliza con las visitas disponibles antes de que el sujeto convierta a MCI, respetando la condición de diseño. Por lo tanto, el %pCU representa la relación entre los pCU detectados en el intervalo de tiempo mencionado respecto del número de pCU real.

Como era de esperar, el AUC y BCA disminuyen para todos los modelos según progresan los años. Lo mismo sucede con el %sCU en GRACE y Leaspy. Por su parte, en RPDPM se mantiene constante. Esto es por la parametrización de los modelos: a medida que avanzan los años tienden a estimar una mayor proporción de conversores. La mayor rigidez en las curvas de RPDPM explican su facilidad para detectar sujetos estables. En la predicción constante el %sCU se mantiene en el 100 %. Esto se debe a que la definición de sCU para el pronóstico coincide con el algoritmo de la predicción constante. El %sMCI se mantiene prácticamente constante en GRACE, RPDPM y Leaspy. A diferencia del %sCU, la predicción constante no alcanza el 100 % porque este grupo también considera sujetos que en su última visita de personalización sean distintos de MCI. En tales casos, la estimación sería incorrecta. El %pCU es 0 en la predicción constante debido a que no es capaz de detectar cambios en el diagnóstico. Tanto en GRACE como en Leaspy se mantiene

¹⁰Nótese que en el caso de los métodos paramétricos se trata del *baseline*, mientras que en las RNN corresponde a la salida de la primera secuencia. En otras palabras, existe un desfase igual a la longitud de secuencia entre ambos métodos.

aproximadamente constante, sufriendo un ligero incremento en los años 4 y 5. RPDPM evoluciona de forma contraria: las predicciones caen en el año 5. Según se ha explicado, esto puede deberse a las curvas monotonamente crecientes de los modelos que tienden a estimar un mayor número de conversores (salvo en RPDPM, que estima conversiones más lentas). Además, cabe decir que los tamaños muestrales en dichos años son bastante menores a los de los primeros años.

El número de sujetos analizados con la LSTM es superior al de los métodos paramétricos, a pesar de haberse eliminado aquellos con menos de 4 visitas. Esto se debe a dos factores. Primero, el criterio para realizar las predicciones difiere entre métodos paramétricos y RNN. Segundo, la LSTM requiere que las visitas estén separadas por un intervalo de 6 meses. Como estas son asíncronas, se generan visitas virtuales para cumplir con el alineamiento temporal¹¹. Al igual que en los métodos paramétricos, tanto el AUC como el BCA se reducen cuanto más alejadas sean las predicciones. El %sCU alcanza niveles similares a la predicción constante. El %sMCI experimenta un descenso apreciable. Una razón posible es la capacidad de la LSTM de detectar sujetos reversores¹². La detección de pCU apenas llega al 10%. La predicción a un año no identifica ningún sujeto conversor. La introducción de visitas virtuales presenta el problema del desconocimiento del diagnóstico real en las mismas. Para solventarlo se ha asumido la conversión en el punto medio entre la primera visita MCI y la anterior CU. Sin embargo, esta solución choca contra el criterio de clasificación de pCU, ya que su número es muy reducido en la predicción a un año¹³. Generalmente, las observaciones se realizan anualmente, produciendo un intercalado entre visitas reales y virtuales¹⁴. Entonces, la visita de la predicción es conocida, mientras que la anterior (i.e. la última de personalización) es virtual y estimada como MCI de acuerdo al criterio establecido. Cabe decir que en la secuencia de entrada se ha incluido el diagnóstico, cuando debería inferirse en la salida. Este podría ser uno de los motivos por los que captura trayectorias estables en vez de progresivas.

Por otro lado, el %pCU (según diagnóstico) detectados por los modelos es estable en Leaspy y creciente en GRACE, siendo muy elevado en ambos casos. Una vez más, los elevados valores se deben a la parametrización de las curvas. Además, se está extendiendo el tiempo de estimación 10 años en el futuro, por lo que es lógico que se estime una mayor cantidad de conversores en ese intervalo. RPDPM vuelve a salirse de la norma, por la rigidez de las curvas. Es esencial complementar estos valores con los datos de correlación; de otra forma, simplemente se estaría diciendo que los sujetos van a convertir sin saber el momento. Tanto la correlación con la edad como con la reserva disminuyen con los años futuros. Lo cual resulta evidente porque cada vez se conocen menos puntos en la evolución de los sujetos. Lo que sí es sorprendente es que GRACE supere a Leaspy en la correlación con la reserva, salvo en el año 4. El motivo podría ser la rigidez en las curvas de Leaspy al progresar más lentamente que en GRACE. Sin embargo, es importante resaltar que los valores de los coeficientes de Pearson no llegan a 0.6 y la población de sujetos es muy inferior a la total al solo considerar sujetos pCU. Esto sugiere que las muestras utilizadas no son representativas. RPDPM mejora notablemente los resultados tanto de Leaspy como GRACE. No obstante, presenta una detección de pCU muy inferior. Dicho

¹¹Por ejemplo, si un sujeto dispone de 2 visitas separadas un año, se añadirá una tercera con valores faltantes correspondiente a los 6 meses.

¹²Referido a sujetos reversores en la predicción (no considerados en este estudio). Es decir, diagnosticados MCI al inicio o final de la personalización, pero CU en la predicción. No deben confundirse con sujetos reversores en la personalización, los cuales se han englobado como sMCI.

¹³No llega a ser 0 porque las predicciones a 1 año agrupan tanto estas como las de 6 meses. De los pocos pCU que existan no son correctamente estimados por la LSTM

¹⁴Recuérdese que el intervalo entre visitas es de 6 meses.

de otra forma, es capaz de determinar el momento de la conversión con mayor exactitud, pero sin saber con certeza si el sujeto va a convertir.

La LSTM detecta menos pCU (según diagnóstico) que cualquier método paramétrico. Además, el tiempo de conversión estimado por la LSTM muestra las correlaciones más bajas con la edad y la reserva. A diferencia de los métodos paramétricos, las LSTM no imponen restricciones a sus curvas, por lo que pueden mantener un diagnóstico indefinidamente en el tiempo sin que haya progresión.

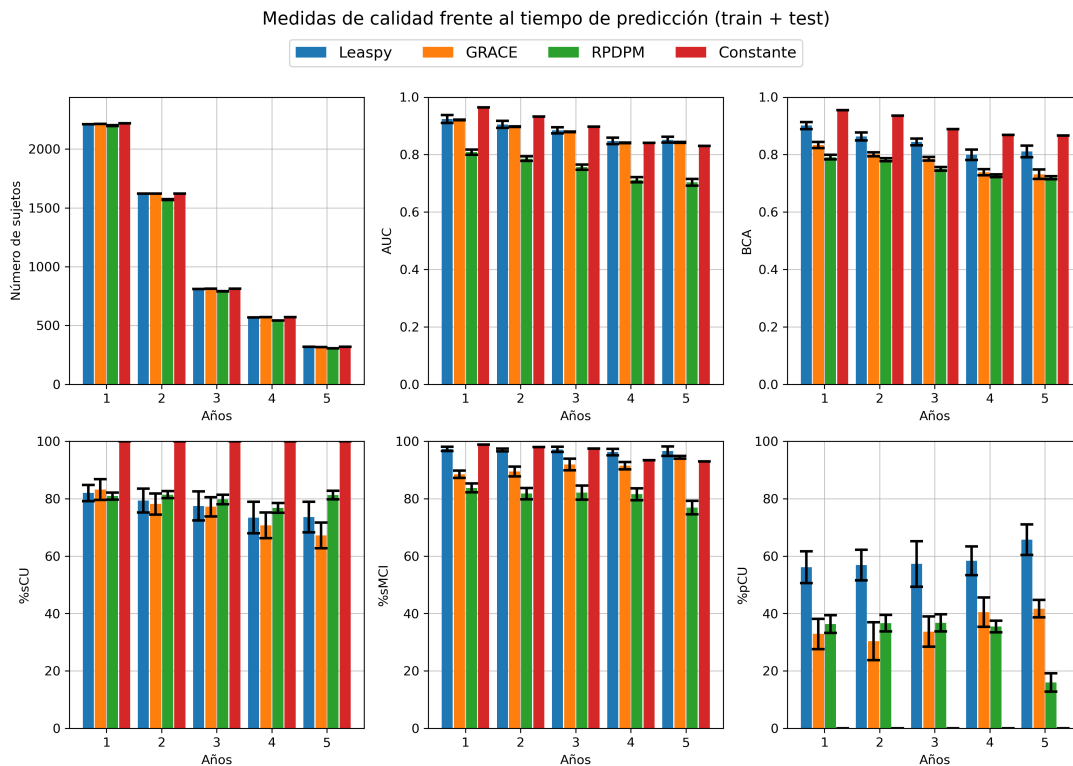


Figura 5.11: Diagramas de barras del número de sujetos, AUC, BCA, %sCU, %sMCI, %pCU frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y MMSE, CDRSB (MC) para GRACE y RPDPM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado combinando los conjuntos de *train* y *test*.

Los errores medios absolutos se presentan en los diagramas de caja y bigotes de las figuras 5.15 y 5.16, donde se han omitido los *outliers*. Debido a que los modelos utilizan distintos marcadores, solamente es posible comparar aquellos que sean comunes. Para los demás, el análisis es únicamente frente a los años futuros y no respecto a otros modelos. La predicción constante puede compararse para cualquier marcador, ya que no requiere entrenamiento con un vector específico.

Para todos los marcadores, la longitud de los bigotes de la predicción constante es menor que en GRACE, RPDPM y Leaspy independientemente del año futuro. No obstante, las medianas siguen una tendencia ascendente cuanto más lejanas en el tiempo sean las predicciones. Leaspy llega a superar a la predicción constante en los años 4 y 5 en ADAS13. GRACE la supera en CDRSB entre los años 3 y 5. RPDPM es más competitivo que GRACE en MMSE, pero menos que la predicción constante. Los cuartiles superiores e inferiores se mantienen aproximadamente constantes en MMSE para la pre-

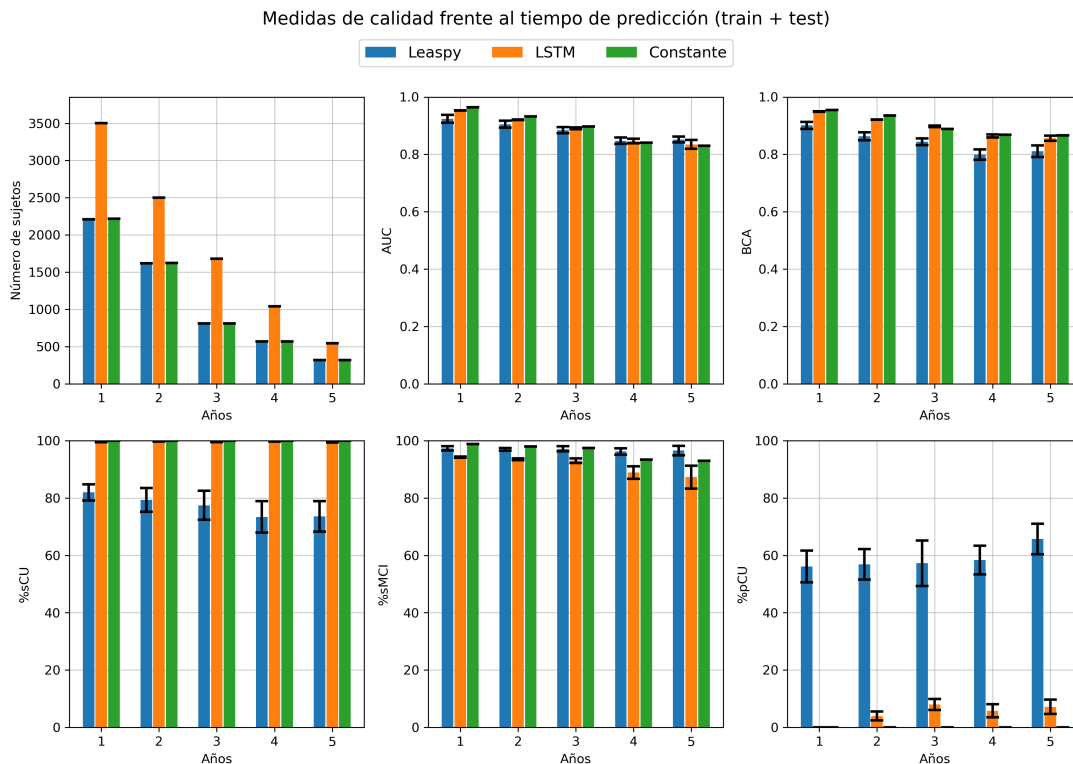


Figura 5.12: Diagramas de barras del número de sujetos, AUC, BCA, %sCU, %sMCI, %pCU frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y RAVLT_learning, FAQ, CDRSB (R_LFC) para LSTM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado combinando los conjuntos de *train* y *test*.

dicción constante, GRACE y RPDPM. En ADAS13, presentan una tendencia creciente, más pronunciada en la predicción constante que en Leaspy. En CDRSB presentan un ligero aumento en todos los modelos, siendo más significativo en la predicción constante a partir del año 3, y en RPDPM desde el primer año.

La longitud de los bigotes de la LSTM es mayor que en la predicción constante, salvo en determinados casos. Las medianas siguen una tendencia creciente. Lo mismo sucede con los cuartiles superior e inferior. Únicamente es posible comparar con Leaspy en CDRSB, donde los errores obtenidos claramente menores.

5.8. Historia natural

Las trayectorias de los pacientes son extendidas 10 años antes y después de la primera y última visita, respectivamente, con una resolución de 0.2 años. En el caso de las RNN, la resolución es de 0.5 años. Esto permite estimar el tiempo de conversión de los sujetos sCU y sMCI. Se usan los modelos GRACE (MC), RPDPM (MC), Leaspy (AC) y LSTM (R_LFC) con el conjunto de toda la población (*train* y *test*).

Las figuras 5.17, 5.18, 5.19 y 5.20 muestran los histogramas con los tiempos de conversión estimados por GRACE, RPDPM, Leaspy y LSTM. Cada figura presenta 4 histogramas, en los que se agrupan los sujetos en:

- sCU, pCU y sMCI: en los métodos paramétricos, el tiempo de conversión es positivo

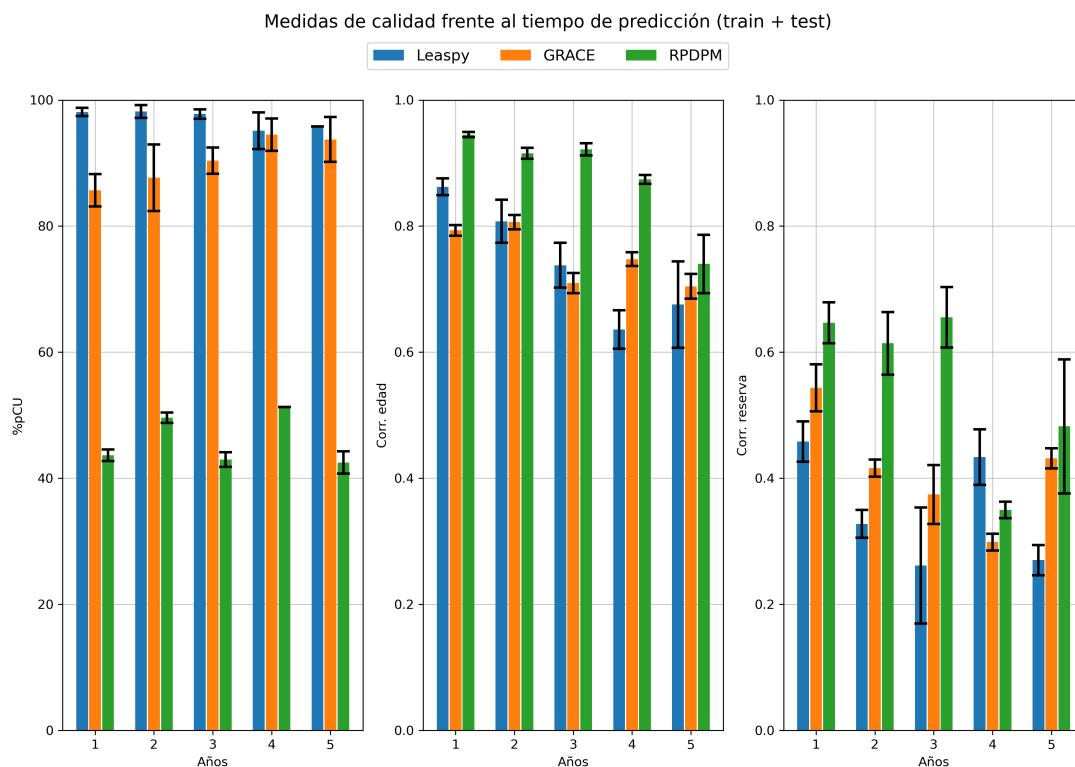


Figura 5.13: Diagramas de barras del %pCU, la correlación con la edad y con la reserva frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y MMSE, CDRSB (MC) para GRACE y RPDPM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado con la población de sujetos pCU disponible en los conjuntos de *train* y *test*.

para los sMCI (ya han convertido) y negativo para los sCU (no han convertido todavía). Los pCU se encuentran en las abscisas negativas, a medio camino entre los sCU y los sMCI. Esta separación presenta un carácter más marcado en Leaspy. En LSTM, la separación entre grupos es más evidente que en Leaspy. No obstante, estima erróneamente una reserva positiva para más de la mitad de los pCU.

- CU y SMC: los SMC se encuentran muy ligeramente desplazados hacia la derecha, es decir, presentan una reserva menor. Se trata de sujetos que todavía no han convertido, por lo que todos los valores positivos corresponden con un error de estimación. Este error es mayor en GRACE y LSTM. RPDPM logra una mayor distinción.
- EMCI y LMCI: presentan una distribución similar con ambos modelos, a pesar de que los LMCI están más próximos a la demencia. Esto es debido a que las visitas con diagnóstico AD fueron eliminadas en el preprocesamiento. Por lo tanto, las visitas están restringidas al mismo intervalo de evolución. En este caso, los valores negativos de reserva son erróneos. De nuevo, GRACE y LSTM cometen un mayor error.
- CU, SMC, EMCI, LMCI: combina las 2 gráficas anteriores. Según se ha comentado, el solapamiento entre sujetos CU y SMC con EMCI y LMCI es mayor en GRACE y LSTM.

Comparando los tiempos de conversión estimados entre algoritmos, se obtienen los gráficos de dispersión de la figura 5.21. La reserva real de los sujetos sCU y sMCI no es conocida. Los primeros no han convertido todavía en las visitas registradas, mientras que los segundos convirtieron antes del *baseline*. Los sMCI cuya conversión es anterior a la

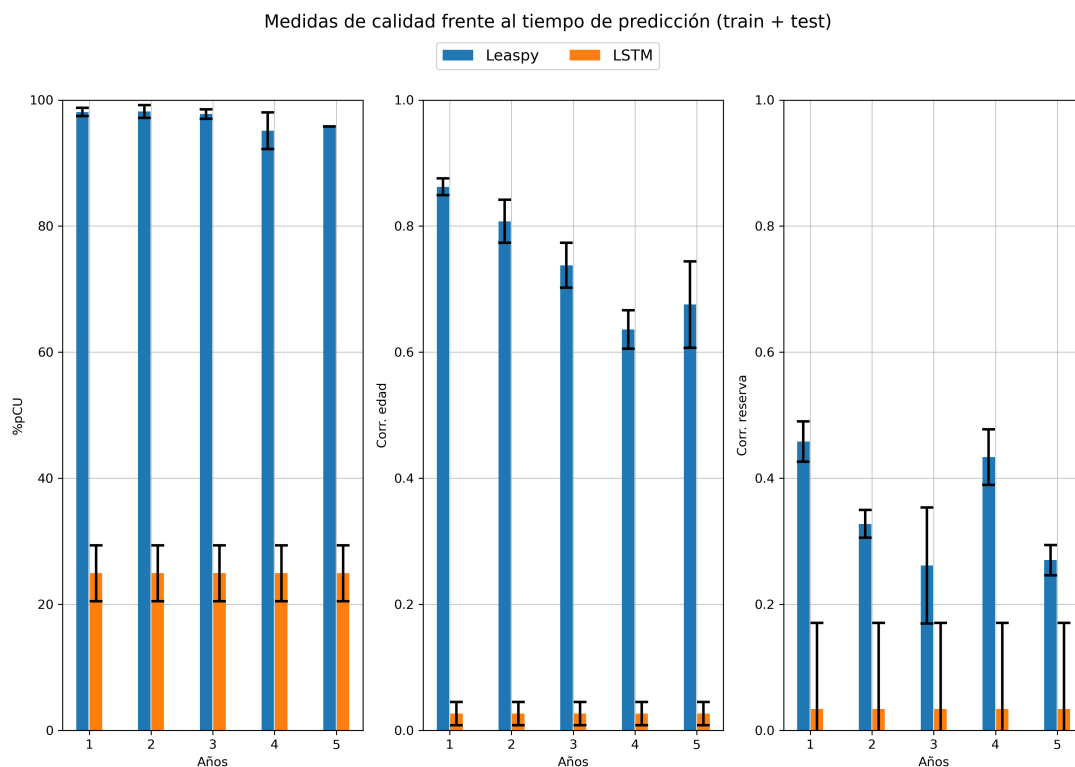


Figura 5.14: Diagramas de barras del %pCU, la correlación con la edad y con la reserva frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y RAVLT.learning, FAQ, CDRSB (R_LFC) para LSTM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado con la población de sujetos pCU disponible en los conjuntos de *train* y *test*.

extensión de la trayectoria se truncan a una reserva 10 años¹⁵. No es posible truncar a los sCU al entenderse que convertirían más allá de los 10 años posteriores a la última visita, pero no 10 años después. Las reservas negativas son superiores en magnitud a las positivas. Esto es porque al período de extensión de las trayectorias se suma el intervalo entre la primera y última visita. La tabla 5.30 muestra las ecuaciones de regresión y los coeficientes de correlación entre los distintos algoritmos, separando por grupo clínico. Debe notarse que los resultados no coinciden con los de la sección 5.4 debido a que se omitieron los errores de clasificación de los algoritmos en la misma. Toda la población pCU se considera en esta sección, aunque no sean detectados correctamente.

Como en otros casos, la recta de regresión ideal es de la forma $y = x + 0$, significando que ambos algoritmos alcanzan el mismo resultado. Las rectas obtenidas siguen las ecuaciones $y = 0,85x + 0,49$ (Leaspy - RPDPM), $y = 0,73x + 0,27$ (Leaspy - GRACE), $y = 0,07x - 0,35$ (Leaspy - LSTM), $y = 0,78x + 0,18$ (RPDPM - GRACE), $y = 0,12x - 0,40$ (RPDPM - LSTM), e $y = 0,05x - 0,36$ (GRACE - LSTM). Los coeficientes de correlación se sitúan en $R = 0.8304$, $R = 0.6821$, $R = 0.0757$, $R = 0.7134$, $R = 0.1342$, y $R = 0.0540$, respectivamente. Entre los métodos paramétricos, las gráficas en las que aparece RPDPM presentan una población sCU más pequeña. La razón, según se comentó en las secciones 5.6 y 5.7, la rigidez en las curvas de RPDPM. Al ser un algoritmo más conservador, la detección de sujetos sCU que convertirán en los próximos 10 años es inferior a Leaspy o GRACE. Ocurre lo mismo con las redes neuronales recurrentes: la poblaciones estables,

¹⁵Para mantener la coherencia con las figuras de esta sección, las reservas positivas deben entenderse como períodos después de la conversión, aunque verbalmente pueda parecer contradictorio.

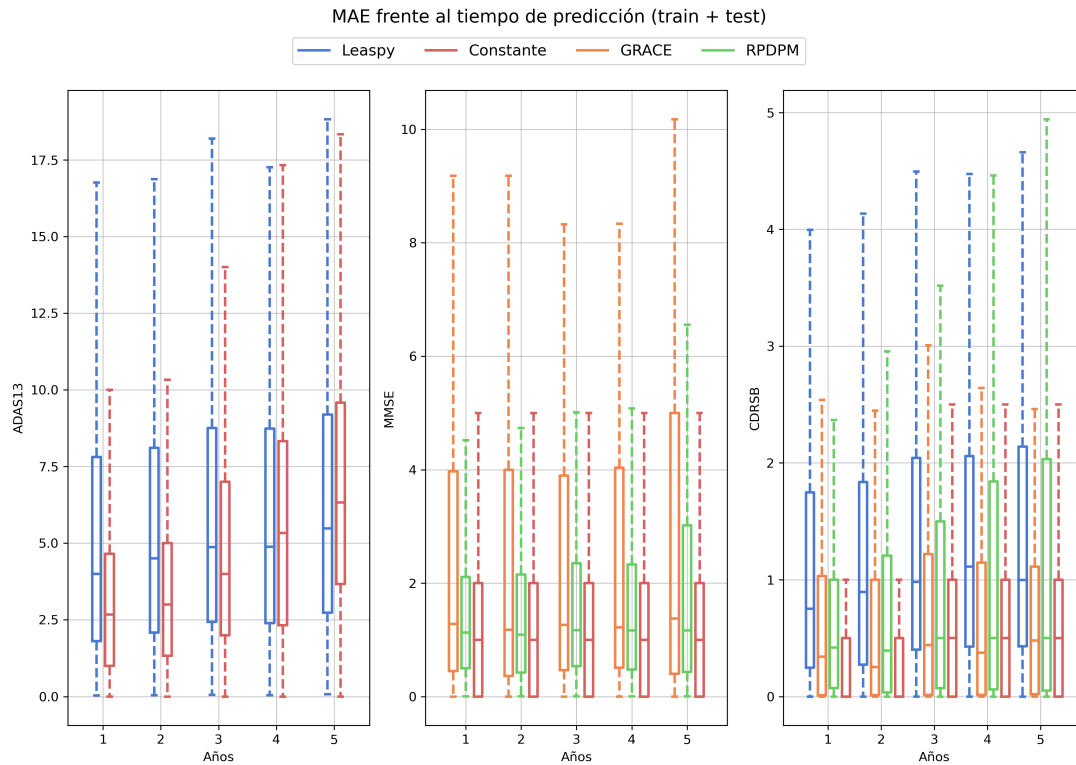


Figura 5.15: Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y MMSE, CDRSB (MC) para GRACE y RPDPM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado combinando los conjuntos de *train* y *test*.

especialmente sCU, son más reducidas porque LSTM puede mantener el diagnóstico indefinidamente. Este argumento queda reforzado al advertir que el tiempo de conversión de los sMCI coincide con el último instante en el pasado¹⁶. Se observa también que GRACE es el que peor correlaciones presenta con Leaspy y RPDPM. Por su parte, LSTM está totalmente incorrelado con los métodos paramétricos.

Atendiendo a la tabla 5.30, los coeficientes de correlación son superiores en sujetos pCU. Esto concuerda con que su conversión se da en el intervalo de las visitas, en vez de explorando períodos de tiempo desconocidos. Los sCU presentan mayores correlaciones que los sMCI debido a que el truncamiento en los tiempos de conversión de estos últimos afecta negativamente.

Estimados los tiempos de conversión de cada sujeto, se ordenan según el punto en la historia natural de la enfermedad en el que se encuentren. Trazando la curva logística propuesta en [38], y optimizando los parámetros p_k , v_k , t_0 se obtienen las gráficas de las figuras 5.22 y 5.23 según GRACE y Leaspy, respectivamente. Los parámetros a optimizar en RPDPM son b , c , γ , de acuerdo con [31], dando lugar a la figura 5.24. La LSTM no tiene una curva prefijada, por lo que se ajusta con *splines*. Al tener el ordenamiento temporal, este procedimiento se puede aplicar a cualquier marcador. Cabe decir que no deben compararse los MAE con este procedimiento con objeto de comparar modelos¹⁷. Esto es

¹⁶Recuérdese que no puede haber sujetos sMCI no estimados. Su conversión se trunca al primer instante disponible en el caso de seguir considerándose estables después de la extensión de trayectorias.

¹⁷Únicamente es posible comparar los MAE con los que fueron entrenados los modelos.

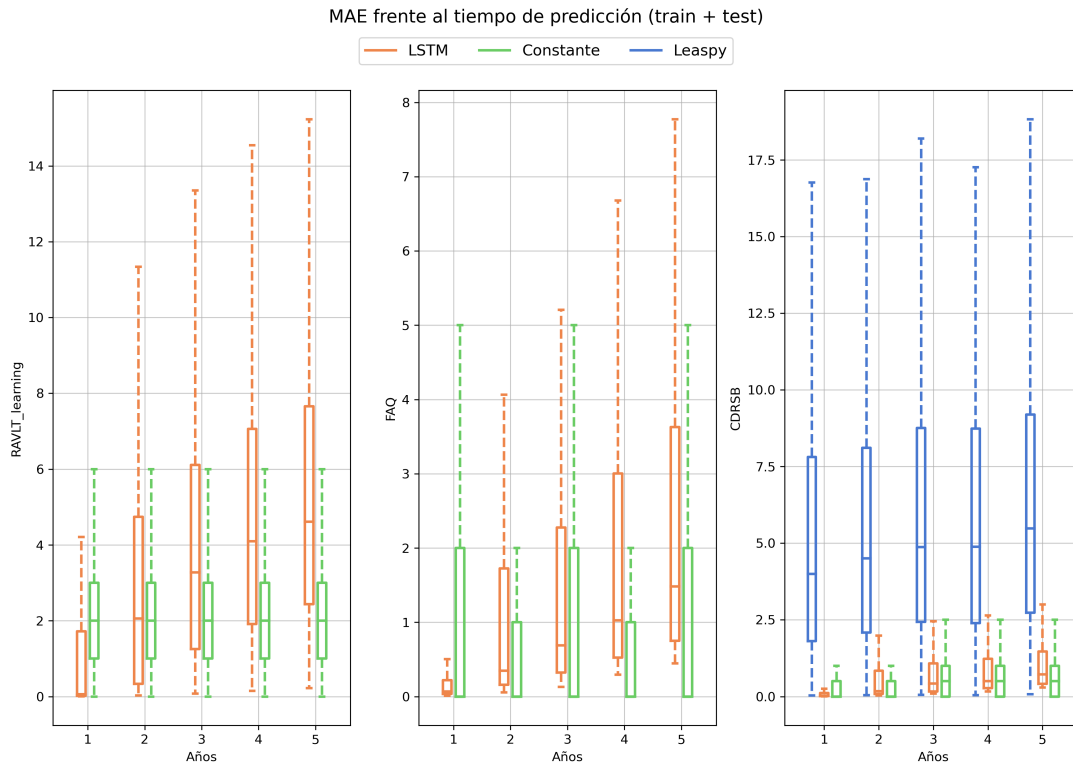


Figura 5.16: Diagramas de caja y bigotes de los errores medios absolutos de cada marcador frente a los años futuros. Los vectores de características empleados son ADAS13, CDRSB (AC) para Leaspy, y RAVLT_learning, FAQ, CDRSB (R_LFC) para LSTM. La predicción constante no requiere entrenamiento (i.e. no hay un mejor vector de características). Los gráficos se han generado combinando los conjuntos de *train* y *test*.

	sCU	pCU	sMCI
Leaspy - RPDPM	Eq. $y = 0.66x + -1.18$ r $r = 0.6515$	$y = 0.78x + -1.34$ r $r = 0.8022$	$y = 0.42x + 3.93$ r $r = 0.3152$
Leaspy - GRACE	Eq. $y = 0.50x + -1.76$ r $r = 0.3248$	$y = 1.00x + -0.46$ r $r = 0.5391$	$y = 0.49x + 2.39$ r $r = 0.2984$
Leaspy - LSTM	Eq. $y = 0.09x + -13.29$ r $r = 0.1227$	$y = 0.13x + -5.06$ r $r = 0.2163$	$y = -0.00x + 3.72$ r $r = -0.0113$
RPDPM - GRACE	Eq. $y = 0.57x + -3.02$ r $r = 0.4872$	$y = 1.45x + 2.34$ r $r = 0.7226$	$y = 0.11x + 5.69$ r $r = 0.0977$
RPDPM - LSTM	Eq. $y = 0.10x + -13.51$ r $r = 0.1744$	$y = 0.04x + -5.27$ r $r = 0.0675$	$y = -0.01x + 3.76$ r $r = -0.0600$
GRACE - LSTM	Eq. $y = -0.00x + -14.21$ r $r = -0.0044$	$y = 0.08x + -5.20$ r $r = 0.2266$	$y = -0.00x + 3.72$ r $r = -0.0248$

Tabla 5.30: Ecuaciones de regresión y coeficientes de correlación con la reserva entre algoritmos, por grupo clínico.

porque los problemas del ajuste espacial y temporal quedan desacoplados, mientras que el entrenamiento de los modelos los resuelven simultáneamente. Un buen modelo es aquel capaz de realizar un ordenamiento temporal correcto cometiendo el menor error espacial posible. En este caso, conocido el ordenamiento temporal optimiza el ajuste espacial. Con lo cual se trata de un problema totalmente distinto.

En las figuras 5.22, 5.24, 5.23 y 5.25 se ha reconstruido la historia natural de ADAS13,

Distribución de los sujetos según el tiempo de conversión (GRACE, train + test)

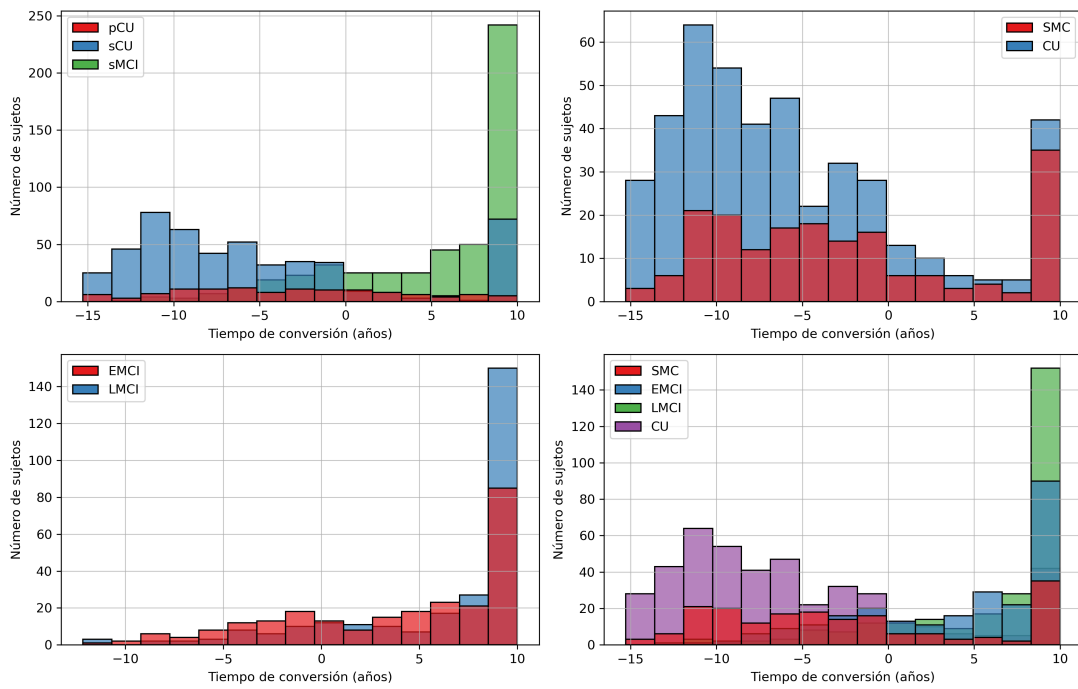


Figura 5.17: Histogramas del número de sujetos frente al tiempo de conversión según GRACE empleando MMSE y CDRSB (MC). Los sujetos se agrupan en sCU, pCU y sMCI; CU y SMC; EMCI y LMCI; CU, SMC, EMCI y LMCI. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

Distribución de los sujetos según el tiempo de conversión (RPDPM, train + test)

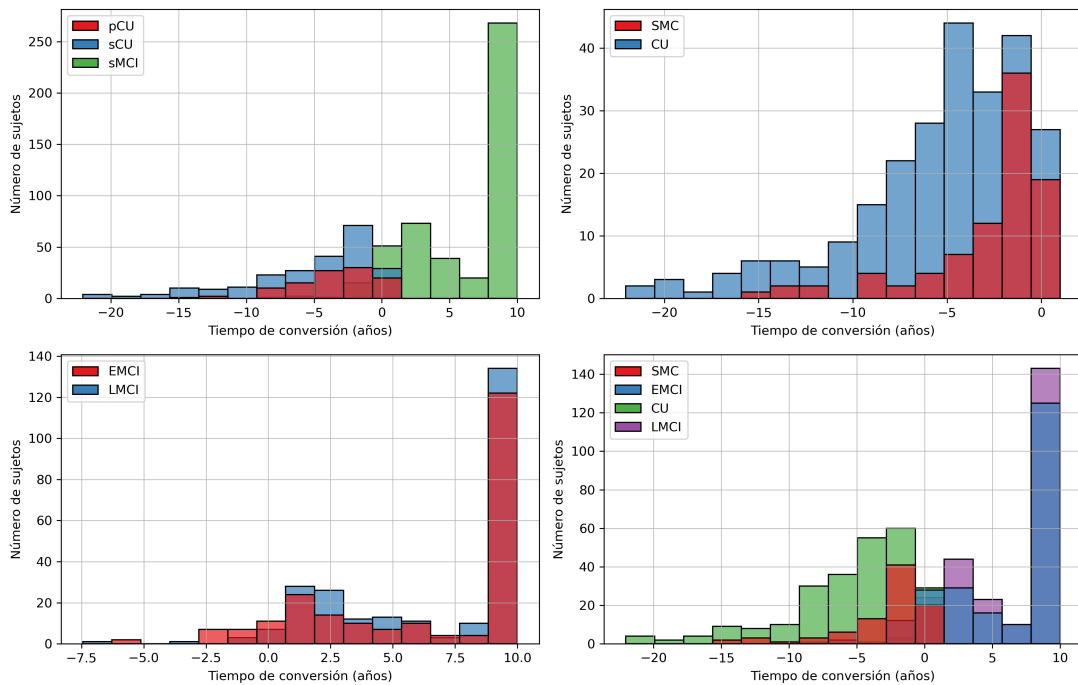


Figura 5.18: Histogramas del número de sujetos frente al tiempo de conversión según RPDPM empleando MMSE y CDRSB (MC). Los sujetos se agrupan en sCU, pCU y sMCI; CU y SMC; EMCI y LMCI; CU, SMC, EMCI y LMCI. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

Distribución de los sujetos según el tiempo de conversión (Leaspy, train + test)

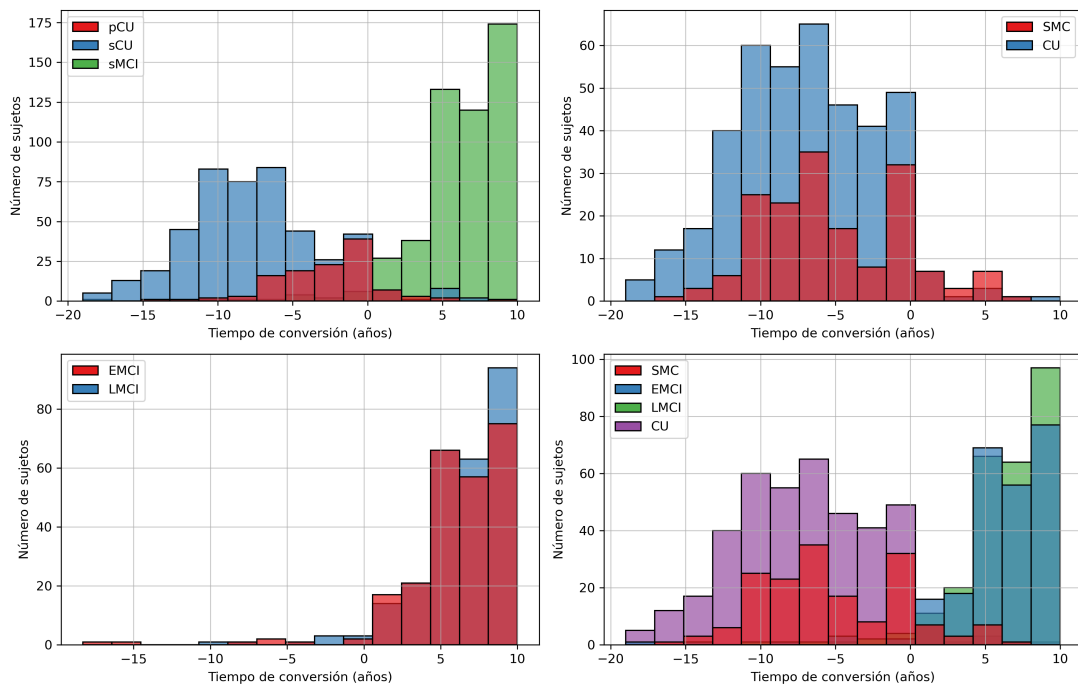


Figura 5.19: Histogramas del número de sujetos frente al tiempo de conversión según Leaspy empleando ADAS13 y CDRSB (AC). Los sujetos se agrupan en sCU, pCU y sMCI; CU y SMC; EMCI y LMCI; CU, SMC, EMCI y LMCI. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

Distribución de los sujetos según el tiempo de conversión (LSTM, train + test)

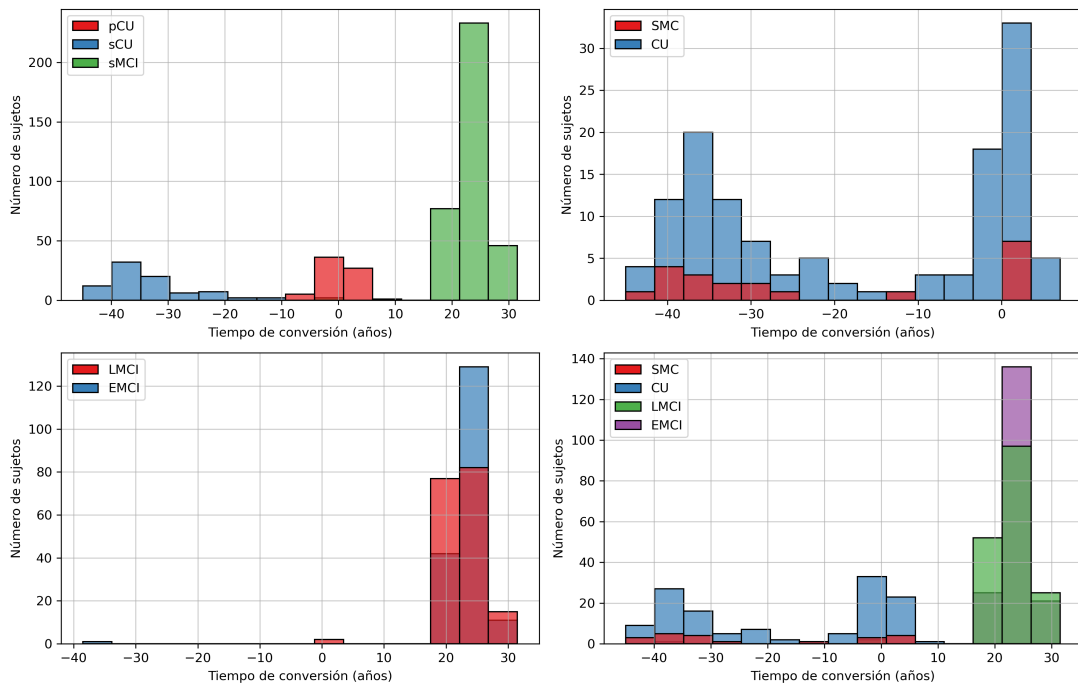


Figura 5.20: Histogramas del número de sujetos frente al tiempo de conversión según LSTM empleando RAVLT learning, FAQ, CDRSB (R_LFC). Los sujetos se agrupan en sCU, pCU y sMCI; CU y SMC; EMCI y LMCI; CU, SMC, EMCI y LMCI. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

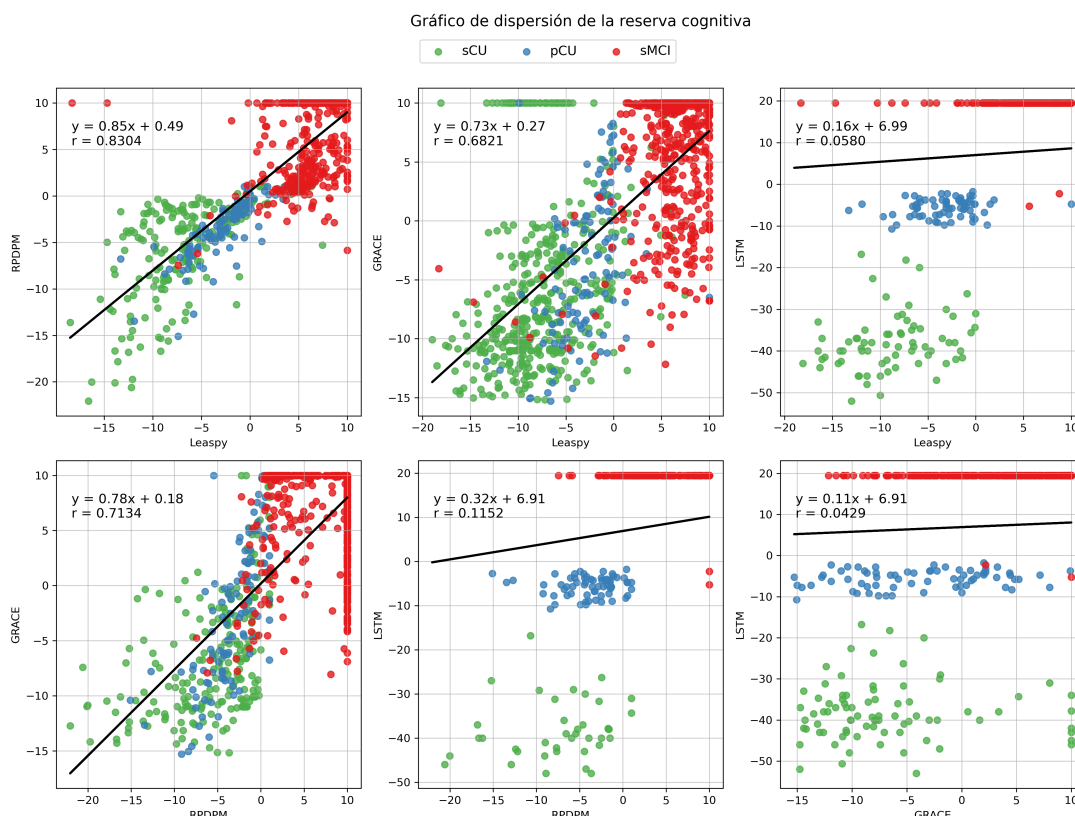


Figura 5.21: Gráficos de dispersión de los tiempos de conversión estimados por Leaspy (AC), RPDPM (MC) y GRACE (MC) de sujetos sCU, pCU y sMCI. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

MMSE, CDRSB, LDELTOTAL, $A\beta$ y NHV^{18 19}. Queda en evidencia que en el ordenamiento temporal propuesto por Leaspy se reduce el solapamiento entre grupos en comparación con GRACE. RPDPM se sitúa a medio camino entre ambos métodos. LSTM separa adecuadamente entre sCU, pCU y sMCI. Las estimaciones de reserva de gran parte de los segundos no se sitúan en el lado correcto de la gráfica. Los últimos presentan un inicio abrupto para una reserva de 10 años. Claramente, las puntuaciones de $A\beta$ y NHV no permiten discriminar entre grupos, ya que hay mucha dispersión de valores en todos los grupos. En este sentido, ADAS13, MMSE y CDRSB logran una mejor distinción.

La tabla 5.31 compara los valores medios de ADAS13, CDRSB y $A\beta$ al inicio del estudio de los sujetos sCU, pCU y sMCI con los valores de las trayectorias a largo plazo cuando el tiempo de conversión es 0. Los valores de corte estimados por GRACE, RPDPM, Leaspy y LSTM son similares a los clínicos en los marcadores NM. Además, se encuentran entre los valores medios de los grupos pCU y sMCI. Los valores de corte estimados para $A\beta$ no se relacionan con los clínicos.

¹⁸Nótese que GRACE y RPDPM han sido entrenados con MMSE, CDRSB (MC), y LSTM con RAVLT Learning, FAQ, CDRSB (R_LFC), mientras que Leaspy con ADAS13, CDRSB (AC).

¹⁹El volumen hipocampal normalizado (NHV) se calcula como el volumen hipocampal dividido por el volumen intracraneal del individuo.

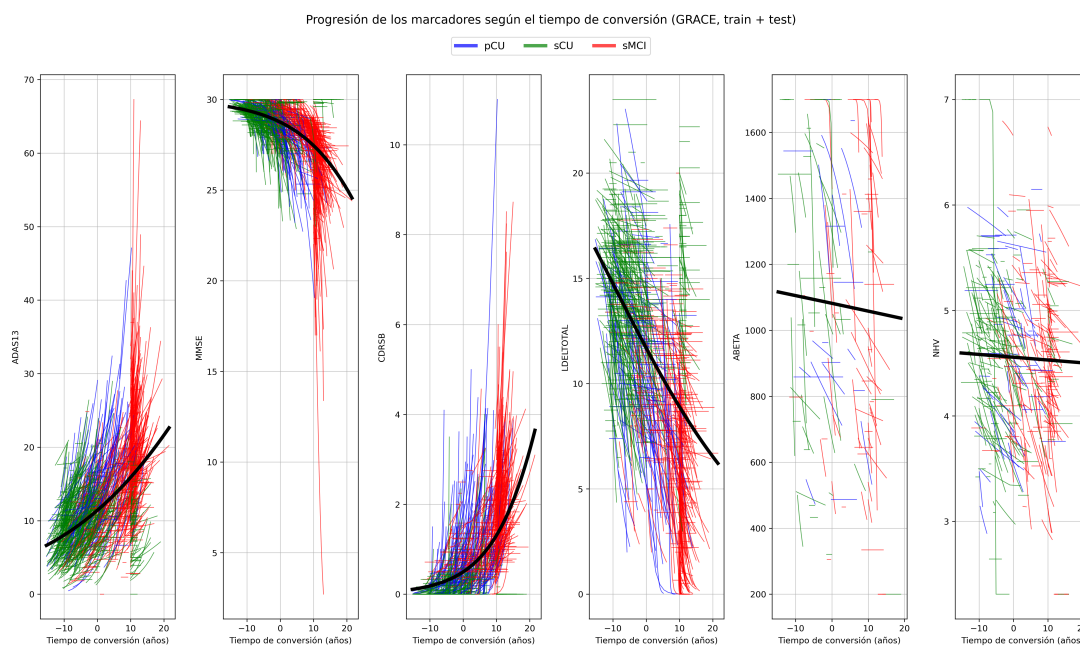


Figura 5.22: Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTOTAL, ABETA, NHV propuesta por GRACE. En verde, azul, rojo las trayectorias a corto plazo de los sujetos sCU, pCU, sMCI, respectivamente. En negro, las trayectorias a largo plazo de los marcadores. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

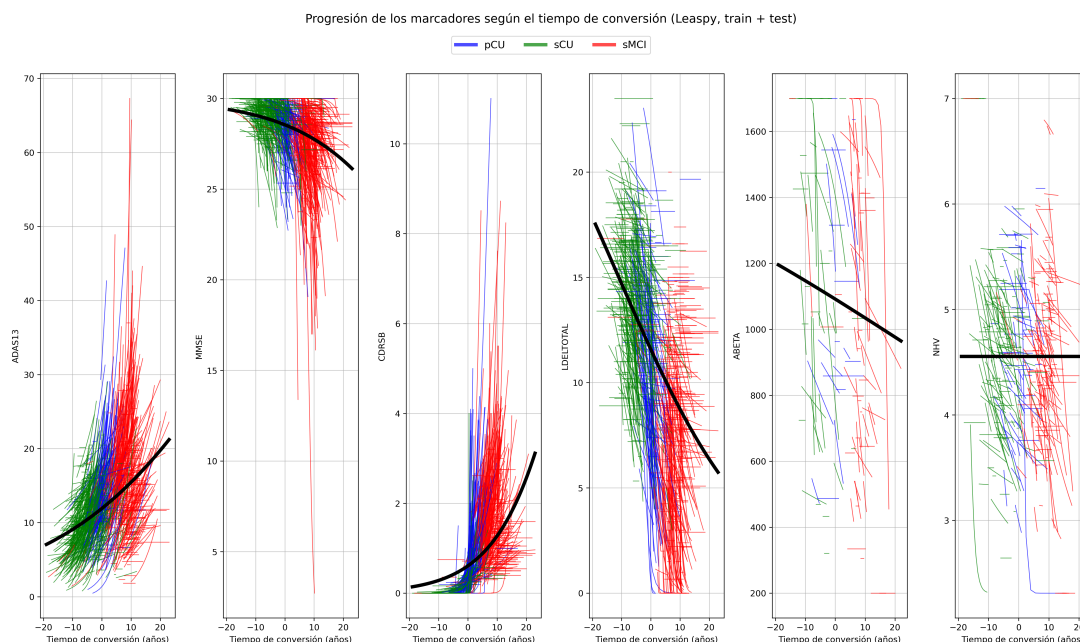


Figura 5.23: Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTOTAL, ABETA, NHV propuesta por Leaspy. En verde, azul, rojo las trayectorias a corto plazo de los sujetos sCU, pCU, sMCI, respectivamente. En negro, las trayectorias a largo plazo de los marcadores. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

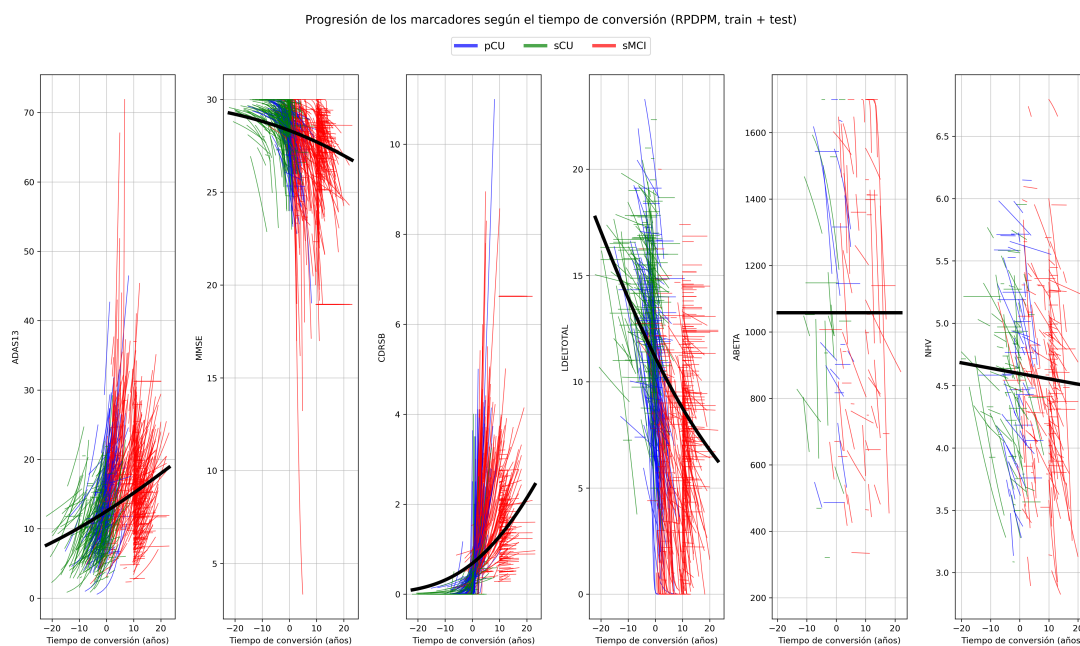


Figura 5.24: Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTOTAL, ABETA, NHV propuesta por RPDPM. En verde, azul, rojo las trayectorias a corto plazo de los sujetos sCU, pCU, sMCI, respectivamente. En negro, las trayectorias a largo plazo de los marcadores. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

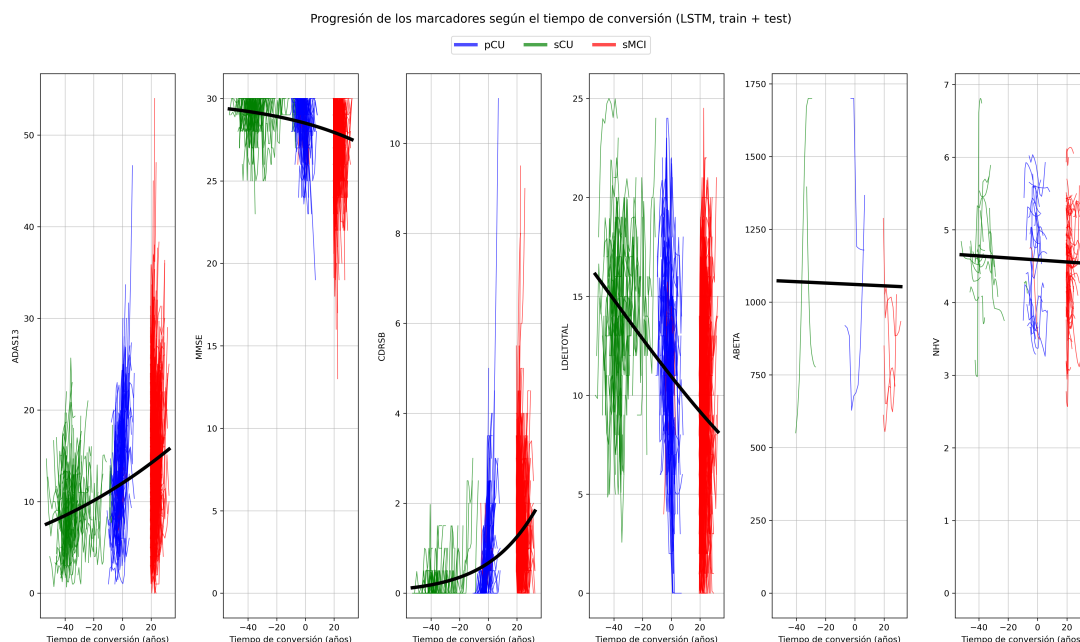


Figura 5.25: Historia natural de los marcadores ADAS13, MMSE, CDRSB, LDELTOTAL, ABETA, NHV propuesta por LSTM. En verde, azul, rojo las trayectorias a corto plazo de los sujetos sCU, pCU, sMCI, respectivamente. En negro, las trayectorias a largo plazo de los marcadores. Los gráficos se han generado combinando los conjuntos de *train* y *test*.

	sCU	pCU	sMCI	clinical cutoff	Leaspy cutoff	GRACE cutoff	RPDPM cutoff	LSTM cutoff
ADAS13	9.57	11.43	15.19	12 [49]	11.92	11.4	12.52	12.29
MMSE	29.09	28.96	27.86	27 [42]	28.54	28.75	28.32	28.38
CDRSB	0.03	0.06	1.3	0.5 [45]	0.61	0.5	0.69	0.85
ABETA	1049.84	1061.47	1013.42	880 [34]	1091.73	1081.84	1058.38	1058.87
NHV	4.52	4.54	4.6	3.8 [27]	4.55	4.56	4.6	4.58

Tabla 5.31: Valores medios de los grupos sCU, pCU y sMCI para las características ADAS13, MMSE, CDRSB, ABETA y NHV al inicio del estudio; valores de corte utilizados en la práctica clínica para el diagnóstico de MCI; valores de corte estimados por Leaspy, GRACE, RPDPM y LSTM.

5.9. Visualización de trayectorias

El estudio busca como objetivo último reconstruir las trayectorias de los sujetos a partir de las muestras. La visualización de la historia natural brinda al personal médico una herramienta en la que apoyarse a la hora de emitir su diagnóstico (o pronóstico).

Las figuras 5.26 y 5.27 muestran una posible salida de la aplicación usando Leaspy (AC) y LSTM (R_LFC) como motores, respectivamente. Cada uno de los marcadores con los que se han entrenado los algoritmos se reconstruyen en su escala original. Cabe decir que, al igual que se hizo en la sección 5.8, la visualización podría extenderse a marcadores distintos de los de entrenamiento. Los puntos representan los valores reales del marcador en cuestión. Las líneas son las trayectorias estimadas por el algoritmo (tantas como se haya fijado en el parámetro de *bootstrap*). La línea negra a trazos indica la edad de conversión estimada por el algoritmo.

Las trayectorias pueden representarse todas en la misma gráfica si se normalizan, dando lugar a las figuras 5.28a y 5.28b. En el caso de las redes neuronales recurrentes se aplica estandarización, por lo que los valores no quedan restringidos entre 0 y 1.

En las figuras 5.26, 5.27, 5.28a y 5.28b se ha representado la evolución durante 25 años de un sujeto progresivo. Claramente, se ve el carácter monotonamente creciente de Leaspy, mientras que LSTM presenta trayectorias más irregulares, sin una forma predefinida.

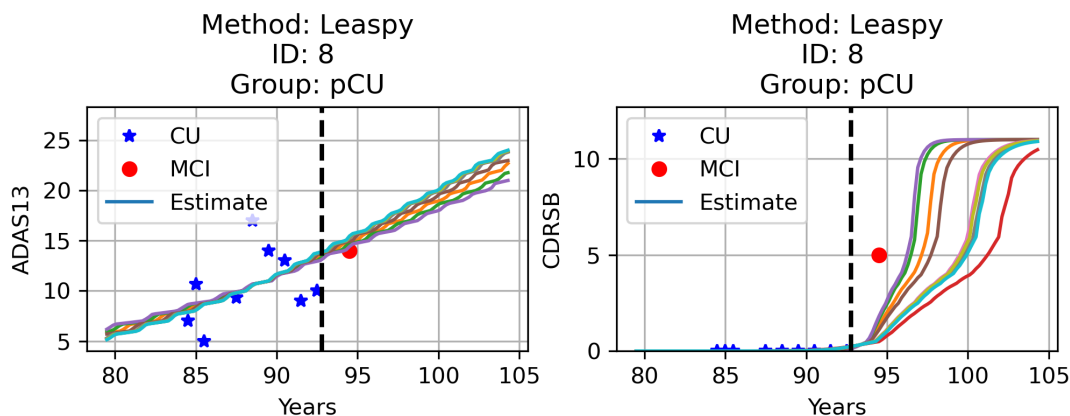


Figura 5.26: Visualización de las trayectorias propuestas por Leaspy (AC) para los marcadores ADAS13, CDRSB en escala original del sujeto con identificador 8. Se tienen tantas curvas como indique el parámetro de *bootstrap*. La línea negra a trazos representa el tiempo estimado de conversión. Los puntos son las muestras reales.

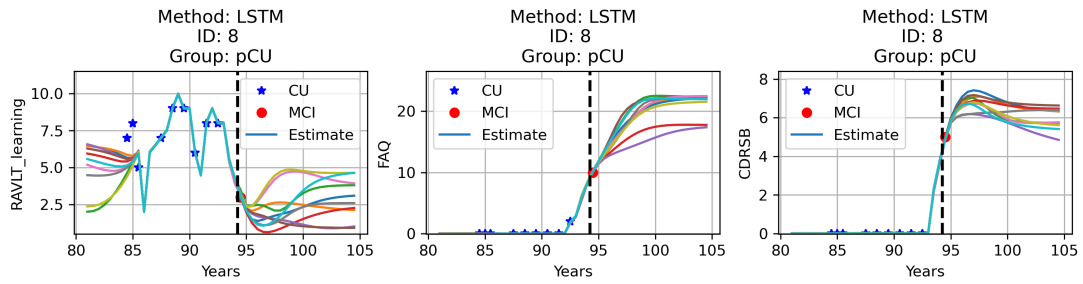


Figura 5.27: Visualización de las trayectorias propuestas por LSTM (R_{LFC}) para los marcadores RAVLT_learning, FAQ, CDRSB en escala original del sujeto con identificador 8. Se tienen tantas curvas como indique el parámetro de *bootstrap*. La línea negra a trazos representa el tiempo estimado de conversión. Los puntos son las muestras reales.

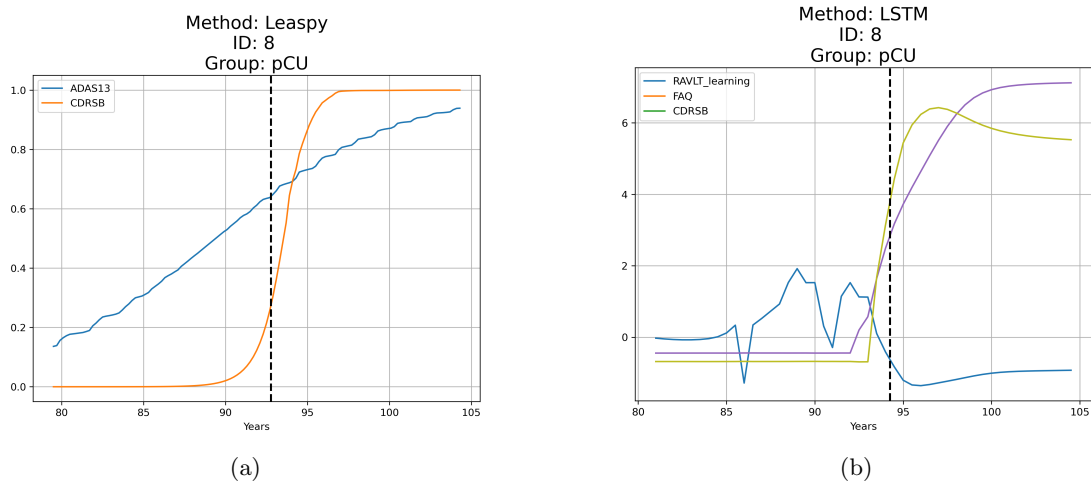


Figura 5.28: Visualización de las trayectorias propuestas por Leaspy (AC) y LSTM (R_{LFC}) para los marcadores ADAS13, CDRSB en escala percentil (a) y RAVLT_learning, FAQ, CDRSB en escala estandarizada (b) del sujeto con identificador 8. La curva resultante es la media de las estimaciones de cada *bootstrap*. La línea negra a trazos representa el tiempo estimado de conversión.

Capítulo 6

Discusiones

Los resultados obtenidos en el capítulo 5 se analizan y comparan con otras investigaciones a continuación.

6.1. Caracterización de la población de estudio

La población de ADNI se clasificó según su grupo clínico al inicio del estudio. Los sujetos MCI resultaron los más numerosos, seguidos por CN, AD y SMC, en el orden indicado. La cohorte se redujo hasta más de 2 veces al requerir medidas CSF, conservando el orden mencionado. Las visitas disponibles se vieron afectadas por factores incluso menores de un sexto¹.

Adicionalmente, se condujo un análisis transversal en el *baseline* entre los sujetos CU y SMC. Los primeros constituyeron alrededor del 60 % de la población y los últimos el 40 % restante. Del total, el 85 % se clasificaron como estables, quedando un 15 % de progresivos. Estos últimos representaron una población ligeramente más envejecida. Los *tests* neuropsicológicos CDRSB, ADAS13 y RAVLT Learning mostraron una mayor capacidad de distinguir entre grupos. Battista et al. [24] reportó que LDELTOTAL, FAQ, RAVLT y ADAS-Cog se encuentran entre los mejores predictores para la clasificación automática. Al incluir medidas CSF el efecto fue similar al observado con la población general (i.e. una reducción significativa de la cantidad de sujetos y visitas).

Entre estos grupos, $\frac{pTAU}{A\beta_{42}}$ mostró ser más discriminativo que $A\beta$, de acuerdo con [34] y [47]. Atendiendo al criterio de $\frac{pTAU}{A\beta_{42}} > 0,028$, la prevalencia de patología amiloide alcanzó un 24 % entre los CU y un 22 % entre los SMC. Este resultado es coherente con el de otros investigadores [44]. Para los conversores, el porcentaje se elevó a un 42 % en pCU y el 38.5 % en pSMC. Entre los conversores rápidos los porcentajes eran de un 41 % en pCU y un 37.5 % en pSMC. En cambio, la proporción de sujetos estables con patología amiloide desciende alrededor del 19.5 % en tanto sCU y sSMC.

Los modelos de progresión de la enfermedad se entrenaron con medidas neuropsicológicas debido a su capacidad de detectar cambios cognitivos y su amplio uso durante el seguimiento en la práctica clínica[47] [29]. De esta forma, se aumenta el conjunto de datos utilizados. Dado que el estudio se centra en la fase preclínica de la enfermedad, los sujetos con demencia fueron eliminados. Lo mismo se hizo con aquellos con menos de dos visitas. La población pCU registró 114 sujetos (un 9.8 % del total) con 795 visitas (12.5 %

¹Entiéndase “menores” como una reducción más notable.

del total), en consonancia con lo obtenido en otros estudios [37]. A pesar de ser la más representativa por conocerse el tiempo de conversión, los sCU y sMCI agruparon un mayor número de sujetos. Algunos *tests* como el ADAS13, LDELTOTAL, PACC o RAVLT² mostraron diferencia significativa entre todos los grupos. Otros (MMSE, CDRSB o FAQ) distinguieron entre CU y MCI, independientemente de ser clasificados además como estables o progresivos.

6.2. Análisis masivo

Los modelos propuestos se separaron en paramétricos (GRACE [28], RPDPM [31] y Leaspy [38]) y no paramétricos (LSTM). El entrenamiento con todas las posibles combinaciones de 2 a 5 características entre: RAVLT (Immediate, Learning, Percent Forgetting, Forgetting), ADAS13, FAQ, MMSE, CDRSB, LDELTOTAL, PACC, TRABSCOR. Donohue et al. [28] y Nguyen et al. [39] llevaron a cabo estudios de ablación después de entrenar con todas las características. Platero et al. [46] usaron modelos predictivos para preseleccionar los mejores DPM.

En todos los casos, las combinaciones óptimas incluyeron más de 3 marcadores. Específicamente, 4 marcadores fueron requeridos por GRACE y RPDPM, y 5 por Leaspy y LSTM. No solo estos últimos demandan una mayor cantidad de datos, sino que en LSTM no se documentó ninguna combinación de 2 o 3 características entre las 5 mejores. Esto es consistente con que las redes neuronales presentan un mayor número de parámetros a actualizar.

Salvo en GRACE, los conglomerados mPACCdigit y mPACCtrailsB se repitieron sistemáticamente entre las mejores combinaciones. Por razones de tiempo y recursos requeridos en la práctica clínica, se seleccionaron vectores de 2 (preferiblemente) o 3 marcadores, excluyendo los composites mencionados, para la construcción de los DPM. En concreto, se utilizó ADAS13 y CDRSB (AC) para Leaspy, MMSE y CDRSB (MC) para GRACE y RPDPM, RAVLT Learning, FAQ, CDRSB (R_LFC) para LSTM.

Se definió el criterio para evaluar la calidad de los algoritmos como la media geométrica del producto de detección de pCU por correlación con la reserva entre los conjuntos de *train* y *test*. La decisión de considerar las métricas del entrenamiento se debió a que los modelos no vieron la totalidad de los datos debido al *bootstrapping*. De los métodos paramétricos, Leaspy obtuvo el mejor desempeño, seguido de cerca por RPDPM y a una gran distancia de GRACE. Aunque RPDPM logró correlaciones con la reserva más fuertes, Leaspy anotó una tasa superior de sujetos pCU detectados. Esto sugiere que la precisión del tiempo de conversión mejora a costa de disminuir el número de sujetos conversores detectados³. Por otro lado, se observó un compromiso entre el ajuste temporal y espacial: los valores de MAE más pequeños fueron arrojados por GRACE, mientras que Leaspy demostró un peor comportamiento.

Las redes neuronales recurrentes, por su parte, quedaron a medio camino entre Leaspy y RPDPM. Alcanzaron las detecciones de pCU más altas entre los métodos propuestos, con correlaciones más débiles de acuerdo a lo expuesto en el párrafo anterior. No obstante, la población de estudio se limitó a sujetos con al menos 4 visitas. Además, las visitas y valores faltantes fueron interpolados linealmente. Este enfoque podría inducir

²Tanto *Immediate*, *Learning* como *Percent Forgetting*.

³La correlación con la reserva se calculó únicamente con aquellos sujetos conversores detectados por los DPM.

un sesgo positivo para la red, ya que implica considerar las observaciones en el instante futuro. Alternativas como la imputación hacia delante (i.e. rellenar con el último valor observado) y autoimputación (i.e. el propio modelo estima los valores faltantes durante el entrenamiento) salvarían el problema. En términos de MAE, la LSTM logró resultados comparables, e incluso superiores, que los de GRACE.

6.3. DPS y tiempos de conversión

Los mejores vectores obtenidos del análisis masivo se emplearon para representar los sujetos según su DPS en el *baseline*⁴ y estimar los tiempos de conversión de los progresivos.

GRACE presentó un mayor solapamiento entre las distribuciones de CU y sMCI que RPDPM y Leaspy, implicando un error de clasificación superior. En todos los casos, los sujetos progresivos se situaron entre medias del DPS de los estables cognitivamente normales y con deterioro cognitivo leve. Asimismo, los sujetos con queja subjetiva quedaron desplazados ligeramente hacia el deterioro cognitivo leve. Este efecto fue especialmente evidente en los pSMC. En cuanto a las estimaciones de tiempo de conversión, Leaspy obtuvo la correlación más fuerte con la reserva, seguido por RPDPM y GRACE, respectivamente. El intercepto de GRACE se alejó notablemente del ideal, dando lugar a un sesgo fijo. Sin embargo, la pendiente no se aproximó más a la unitaria que RPDPM y Leaspy. Al comparar los tiempos de conversión entre pares de modelos, Leaspy y RPDPM mostraron mayor consenso que el resto de parejas.

6.4. Robustez frente a valores faltantes

La robustez de los modelos se evaluó introduciendo de forma aleatoria valores faltantes en proporciones de 10 %, 20 % y 40 % sobre el total de observaciones por marcador. Como era de esperar, en todos los casos se observó un descenso en el rendimiento de los algoritmos.

Entre los métodos paramétricos, Leaspy resultó más robusto que RPDPM en términos absolutos. Opuestamente ocurrió en términos relativos. GRACE se vio afectado en mayor medida. Se evidenciaron deficiencias especialmente en la detección de pCU, mientras que el impacto en la correlación con la reserva no fue tan negativo. Los MAE se mantuvieron constantes en Leaspy y disminuyeron en GRACE y RPDPM. La razón ante este comportamiento posiblemente es que se consigue un mejor ajuste con cuantos menos puntos se dispongan.

Las redes neuronales recurrentes experimentaron un menor deterioro, alzándose como el algoritmo más robusto. Hasta el 40 % de valores faltantes artificiales las métricas permanecieron constantes. Solamente el conjunto de *test* registró un notable detrimento en tanto la correlación con la reserva como la detección de pCU. Las desviaciones espaciales se atenuaron. De nuevo, la imputación por interpolación podría estar detrás de los resultados obtenidos. Al existir mayor cantidad de valores faltantes, las predicciones de la red tienden a la visita próxima con la que se ha realizado la interpolación.

⁴Únicamente es posible representar los histogramas con el DPS para los métodos paramétricos. Las redes neuronales recurrentes calculan directamente el diagnóstico sin pasar por una escala temporal intermedia.

6.5. Análisis frente al número de visitas

El análisis en función del número de visitas disponibles por paciente se realizó solamente para los métodos paramétricos. El único requisito para la construcción de redes neuronales recurrentes es que los sujetos dispongan de tantas visitas como longitud tengan las secuencias empleadas. La variación en el número de visitas no altera la longitud de la secuencia por lo que no resulta de interés este análisis.

Leaspy y RPDPM superaron a GRACE en casi todas las puntuaciones analizadas. La detección de sujetos estables sufrió un descenso con el número de visitas, debido a que aumenta la cantidad de puntos con el mismo diagnóstico a situar en la misma zona de la curva. Lo contrario sucedió con los pCU por la razón comentada: las curvas monotonamente crecientes tienden a estimar MCI en la última visita cuanto más tiempo haya pasado desde el *baseline*. Tanto Leaspy como RPDPM fueron capaces de estimar con precisión el tiempo de conversión de sujetos con muchas visitas. Aunque la correlación continuó siendo elevada en los sujetos con menos visitas, su detección experimentó un descenso. El rendimiento de GRACE se alejó considerablemente.

Respecto a los MAE, solamente pudieron compararse aquellos marcadores comunes entre modelos. En general, GRACE y RPDPM presentaron un error espacial menor que Leaspy, creciente con el número de visitas. La explicación que se encuentra reside en la mayor facilidad al ajustar curvas con menos puntos. Destacaron los resultados de GRACE para 2 visitas, logrando anular el error. Durante la personalización, la recta que minimiza el error cuadrático pasa por los únicos puntos disponibles.

6.6. Pronóstico

Los mismos modelos procedentes del análisis masivo se emplearon en el pronóstico de próximas visitas. Eliminando progresivamente observaciones, se utilizaron el resto para extrapolar el comportamiento futuro. Se redefinieron los grupos en función del diagnóstico primera y última visita de la personalización, así como en la visita a predecir. Se introdujo un modelo adicional: la predicción constante. Su funcionamiento consistió en repetir el último diagnóstico observado. Adicionalmente, se trató de predecir la conversión disponiendo únicamente de las visitas cognitivamente normales anteriores al evento.

Los resultados obtenidos con los métodos paramétricos se asemejaron al análisis según el número de visitas. Se produjo una caída en el rendimiento de los sujetos estables al aumentar la lejanía temporal de las predicciones. Lo contrario sucedió con los progresivos. GRACE y Leaspy siguieron fielmente esta tendencia, mientras que RPDPM arrojó unas predicciones más estables. Como era de esperar, la predicción constante registró un 0% en la detección de pCU. Sus resultados en sujetos estables superaron al resto de modelos. Leaspy y GRACE lograron acercarse a estos valores. RPDPM quedó más distanciado.

Las redes neuronales recurrentes sobresalieron en la detección de sCU, superando a Leaspy con creces e igualando a la predicción constante. El porcentaje de sujetos clasificados correctamente como sMCI alcanzó niveles elevados, aunque por debajo de Leaspy y la predicción constante. La calidad de las predicciones en sujetos estables disminuyó gradualmente con la lejanía de las visitas. Los sujetos progresivos no fueron detectados con éxito. Entre las posibles razones figuran la inclusión del diagnóstico en las secuencias de entrada y la falta de restricciones en la forma de las curvas.

Al emplear solo visitas CU, tanto GRACE como Leaspy detectaron casi la totalidad de los sujetos progresivos. RPDPM adoptó una postura más conservadora, aunque con correlaciones más fuertes para los pacientes detectados. La mayor rigidez de la parametrización de las curvas de RPDPM podría dar explicación a estos comportamientos. En cualquier caso, queda reflejado un compromiso entre detección y precisión temporal en la conversión.

Las redes neuronales recurrentes detectaron porcentajes muy reducidos de sujetos progresivos. La flexibilidad de sus curvas permite que no haya una progresión a medida que pasa los años. Entre los sujetos correctamente identificados las correlaciones con los tiempos de conversión se mantuvieron bajas.

Respecto a los MAE, la predicción constante superó a los métodos paramétricos, si bien en puntos más alejados estos llegaron a ofrecer un mayor ajuste. RPDPM logró errores menores que GRACE. Leaspy mostró el peor desempeño espacial. Las redes neuronales recurrentes rebasaron fácilmente a Leaspy, e incluso resultaron competitivas frente a la predicción constante.

6.7. Historia natural

Con los modelos obtenidos se extendieron las trayectorias de los sujetos sCU y sMCI hacia el futuro y pasado, respectivamente. De esta forma, se pudo estimar el tiempo de conversión de pacientes de los que no se conocía a priori. Situando las visitas en el eje de la reserva cognitiva se reconstruyeron las trayectorias tanto individuales como de la población. Esto permitió validar las hipótesis de la evolución biológica del Alzheimer propuestas en [35]. Adicionalmente, se compararon las *pathological scores* calculadas por los modelos y las empleadas en la práctica clínica.

Leaspy dio lugar a unas distribuciones con menos solapamiento entre grupos clínicos que GRACE y RPDPM. Sin embargo, este último logró una mayor distinción entre sujetos CU y SMC. En el resto de casos analizados, el solapamiento en GRACE fue más notable. Al comparar los tiempos de conversión por pares de algoritmos, Leaspy y RPDPM estuvieron en mayor sintonía frente al resto, especialmente en sujetos pCU.

La LSTM no fue capaz de detectar el tiempo de conversión de sujetos estables, por lo que se descartaron algunos sCU y se truncaron muchos sMCI. Aunque la separación entre sCU, pCU y sMCI la realizó adecuadamente, estimó una reserva positiva en gran parte de los segundos. No logró ninguna distinción entre sujetos CU y SMC. Lo mismo ocurrió con EMCI y LMCI. Los tiempos de conversión estimados estuvieron incorrelados con cualquier método paramétrico.

La reconstrucción de las trayectorias no solo fue posible para los marcadores del entrenamiento, sino también para pruebas CSF o T1-MRI. Los métodos paramétricos distinguieron con facilidad los grupos sCU, pCU y sMCI, principalmente en Leaspy. La LSTM tuvo problemas en la estimación de la conversión de sujetos estables. Se observó un corte abrupto en la reserva inicial de los sMCI. Los *tests* neuropsicológicos, en particular MMSE y CDRSB, mostraron una trayectoria en mayor concordancia con la de la población que el $A\beta$ o NHV. Estos últimos presentaron una mayor variabilidad entre grupos, así como trayectorias más dispersas. Esto sugiere el uso exclusivo de medidas neuropsicológicas en

la construcción de DPM para caracterizar el deterioro cognitivo.

Los valores umbrales de conversión a MCI (*pathological scores*) de *tests* neuropsicológicos estimados estuvieron en línea con los usados en la práctica clínica [49], [42], [45]. Por el contrario, las pruebas de otras modalidades difirieron en mayor medida [34], [27].

Capítulo 7

Conclusiones

Se presentan a continuación las conclusiones del estudio realizado, así como líneas de investigación futuras.

7.1. Conclusión

La enfermedad de Alzheimer afecta a la memoria y las funciones cognitivas de quienes la padecen. El impacto no solo es psicológico sobre el paciente y sus familiares, sino que también supone una importante carga financiera. La prevalencia presenta una tendencia creciente, lo que eleva la magnitud del problema. A pesar de que no existe una cura, es posible ralentizar su evolución. Es por ello que se vuelve de vital importancia realizar una detección temprana.

Este estudio ha tenido por objetivo desarrollar un marco de trabajo sobre el que comparar el rendimiento diversos modelos de progresión de la enfermedad. Aunque el análisis se ha restringido a un conjunto concreto de algoritmos, este marco es extensible a nuevos enfoques de futuros trabajos. Las técnicas propuestas se han clasificado en dos grupos: paramétricas (GRACE, Leaspy, RPDPM) y no paramétricas (redes neuronales recurrentes). A continuación, se ha realizado una búsqueda masiva de los marcadores que mejor capturan la evolución de la enfermedad en un entorno de cómputo intensivo. El proceso se ha limitado a *tests* neuropsicológicos, por tratarse de las pruebas utilizadas en la práctica clínica. Además de ser menos invasivas que otras (por ejemplo, punción lumbar), disminuyen los tiempos y costes de implementación. Las combinaciones de marcadores de menor dimensión se han priorizado por la misma razón. Los mejores modelos obtenidos se han sometido a pruebas de robustez, evaluación frente al número de visitas, pronóstico y reconstrucción de la historia natural.

Los resultados han revelado unas puntuaciones comparables a otras investigaciones en materia de diagnóstico. Es más, dichas puntuaciones se han mantenido competitivas al introducir porcentajes elevados de valores faltantes. Se ha verificado que al aumentar el número de visitas, las estimaciones se vuelven más confiables. En relación al pronóstico, se ha determinado no solo la conversión de los sujetos sino también el instante en el que lo harán. Como era de esperar, las predicciones han empeorado conforme más lejanas estuviesen. Al explorar tanto el pasado como el futuro de los sujetos se han podido colocar sobre la línea de evolución de la enfermedad. Esto ha permitido reconstruir la trayectoria de la población para cada marcador. A diferencia de los marcadores de punción lumbar, el umbral patológico de las pruebas neuropsicológicas se ha aproximado a los valores empleados en la práctica clínica.

En conclusión, los modelos desarrollados se anticipan a la evolución de la enfermedad, incluso previamente a la aparición de los síntomas. Esto brinda una valiosa herramienta al personal médico, quienes recibirán apoyo en la toma de decisiones orientadas a ralentizar su transcurso.

7.2. Limitaciones

La principal limitación del estudio es la cohorte utilizada. Los sujetos proceden únicamente de la base de datos de ADNI, regida por criterios de inclusión y exclusión. Se exige que los participantes tengan entre 55 y 90 años, presenten un buen estado de salud y dispongan de una persona de confianza que pueda aportar información sobre los mismos. Se descartan aquellos pacientes con alguna enfermedad neurológica distinta de Alzheimer, condiciones médicas inestables o portadores de dispositivos metálicos (por ejemplo, marcapasos). También se excluyen personas con un historial de abuso de alcohol o drogas superior a 10 años, trastornos psiquiátricos (depresión, bipolaridad o esquizofrenia) que dificulten la evaluación cognitiva, o alteraciones clínicas en la vitamina B12, sífilis o función tiroidea. Por otro lado, la mayor parte de la población de ADNI es caucásica con un nivel elevado de estudios.

Las medidas recogidas por ADNI a menudo presentan irregularidades y valores faltantes. Además, el diagnóstico clínico puede verse afectado por la subjetividad del personal médico. Mientras que la frontera con la demencia es un umbral relativamente establecido, la transición entre el estado cognitivamente normal y el deterioro cognitivo leve puede variar en mayor medida entre centros.

A pesar de la alta calidad de los datos proporcionados por ADNI, las limitaciones mencionadas podrían provocar errores de generalización a otras cohortes y descensos en el rendimiento de los modelos.

7.3. Desarrollos futuros

A continuación, se proponen varias líneas de mejora y ampliación del proyecto:

- Ampliación de los datos a otras cohortes para minimizar el error de generalización.
- Análisis del riesgo según el perfil AT(N), factores genéticos, problemas cardiovasculares, sexo o estatus socioeconómico en la progresión de la enfermedad.
- Exploración alternativas no invasivas como marcadores en sangre [43] capaces de reemplazar a la punción lumbar.
- Desarrollo de un módulo de fusión de modalidades (marcadores estáticos, *tests* neuropsicológicos, medidas de resonancia magnética, punción lumbar...) [50] que proporcione distintos puntos de vista de la evolución de la enfermedad.
- Búsqueda de los hiperparámetros óptimos de las redes neuronales recurrentes. No solamente el número de capas ocultas y neuronas por capa, sino también la ponderación entre variables continuas y categóricas en la función de pérdidas, la longitud de la secuencia o el paso temporal.

- Incorporación de relaciones multivariantes y temporales entre observaciones, de acuerdo con la propuesta de [36].
- Imputación de valores faltantes de forma dinámica durante el entrenamiento.

Apéndice A

Estadística

A.1. Inferencia estadística

La inferencia estadística es una rama de la estadística que busca obtener parámetros de una población a partir de los datos de una muestra. Los parámetros muestrales que estiman los poblacionales se denominan estadísticos. Son una función de n variables aleatorias, con n el tamaño de la muestra.

A menudo la estimación no es puntual, sino por intervalos. Es decir, existe una región dentro de la cual se encuentra el parámetro poblacional en cuestión con una determinada confianza (nivel de confianza).

A.1.1. Contrastes de hipótesis

Los contrastes de hipótesis se utilizan para tomar decisiones acerca de una población en base a la información de una muestra. Se definen:

- Hipótesis nula (H_0): afirmación que se cuestiona y se mantendrá como cierta hasta que los datos evidencien lo contrario.
- Hipótesis alternativa (H_1): afirmación que se aceptará como verdadera cuando se demuestre la falsedad de la hipótesis nula.

Al realizar un contraste de hipótesis se pueden tener errores:

- Error de tipo I: rechaza la hipótesis nula cuando es cierta. La probabilidad máxima aceptable de cometer un error de este tipo se conoce como nivel de significancia.
- Error de tipo II: acepta la hipótesis nula cuando es falsa.

p-valor

El p-valor es la probabilidad del valor del estadístico en cuestión en la dirección de la hipótesis alternativa y partiendo desde el valor real, suponiendo cierta la hipótesis nula.

A continuación, se detalla el procedimiento:

1. Se plantean H_0 y H_1 . Se establecen niveles de significación (α).
2. Se toma una muestra y se calcula el estadístico correspondiente.
3. Se halla la distribución de probabilidad del estadístico correspondiente al parámetro en cuestión.

4. Se calcula la probabilidad del estadístico en cuestión en la dirección de H_1 . Esta probabilidad es el p-valor.
5. Pueden darse 2 situaciones:
 - El p-valor es menor que el nivel de significancia: se rechaza H_0 en favor de H_1 .
 - El p-valor es mayor que el nivel de significancia: se acepta H_0 en favor de H_1 .

Un p-valor pequeño quiere decir que al seleccionar una muestra aleatoria se ha obtenido un resultado considerado improbable. Por el contrario, un p-valor grande indica que el resultado de analizar la muestra aleatoria está dentro de lo esperado.

A.1.2. Prueba de chi cuadrado

Se trata de un test estadístico que se aplica cuando todas las variables del estudio son categóricas. Existen 2 tipos:

- Prueba de chi cuadrado para la bondad de ajuste: se desea averiguar si una variable categórica sigue una determinada distribución. La hipótesis nula es siempre que la variable se ajusta a la distribución en cuestión.
- Prueba de chi cuadrado de independencia: se desea averiguar si hay una relación entre 2 o más variables categóricas. La hipótesis nula es siempre que las variables son independientes.

El procedimiento es el siguiente:

1. Plantear las hipótesis nula y alternativa. Determinar el nivel de significación.
2. Determinar los valores observados (valores de la muestra) y esperados (acogiéndose a lo establecido por H_0).
3. Determinar el valor del estadístico χ^2 según la fórmula:

$$\chi_0^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

4. Usando la distribución χ^2 determinar el p-valor asociado a χ_0^2 .
5. Rechazar/aceptar H_0 según lo expuesto anteriormente.

A.1.3. ANOVA (1 factor)

Se trata de un test estadístico que se aplica cuando al menos una de las variables del estudio es categórica y una es continua dependiente. Se usa para comprobar si existe una diferencia significativa entre las medias de 3 o más grupos. La hipótesis nula es siempre que las medias de los grupos son iguales.

El procedimiento es similar al realizado con la prueba de chi cuadrado. Sin embargo, en este caso se calcula el valor del estadístico F :

$$F_0 = \frac{\widehat{S}_b^2}{\widehat{S}_w^2}$$

Donde:

- La distribución F de Fisher tiene $(a - 1)$, $(n - a)$ grados de libertad.
- $\widehat{S}_b^2 = \frac{v_b}{a-1}$
- $\widehat{S}_w^2 = \frac{v_w}{n-a}$
- $v_b = \sum_j n_j (\bar{x}_j - \bar{x})^2$
- $v_w = v - v_b$
- $v = \sum_{j,k} (x_{jk} - \bar{x})^2$
- a es el número de grupos.
- n es el número total de medidas realizadas. En el caso de que en cada grupo se tome el mismo número de medidas $n = ab$, con b el número de medidas para cada grupo.
- j se refiere a un grupo específico. Así, la notación x_j indica la suma de todas las medidas del grupo j .
- k se refiere a una medida específica.

Se halla el p-valor para F_0 y se compara con el nivel de significación. Con el resultado, se decide si se rechaza/mantiene la hipótesis nula.

Pruebas post-hoc

Se realizan sobre todos los posibles pares de grupos después de rechazar la hipótesis nula (i.e. hay una diferencia significativa entre las medias). De esta forma, se consigue averiguar entre qué grupos existe una diferencia significativa.

Al realizar múltiples comparaciones aumenta el FWER (*Family-Wise Error Rate*). La probabilidad de no equivocarse al realizar un contraste de hipótesis viene determinada por el nivel de significación (p.ej. 0.95 si $\alpha = 0,05$). Si se realizan n contrastes con el mismo α , la probabilidad de no equivocarse en ninguno es $(1 - \alpha)^n$. Por lo tanto, al realizar n contrastes la probabilidad de tener un error de tipo I en alguno de ellos viene dada por $FWER = 1 - (1 - \alpha)^n$.

Por ejemplo, si se aplica un tratamiento post-hoc a 4 grupos (A, B, C, D), se tendrán que realizar 6 comparaciones entre grupos (AB, AC, AD, BC, BD, CD). En este caso, suponiendo $\alpha = 0,05$, se tiene que $FWER = 1 - (1 - 0,05)^6 = 0,265$. La probabilidad de tener un error de tipo I aumenta considerablemente.

Método de Bonferroni

El método de Bonferroni surge para dar solución al problema de realizar comparaciones múltiples (existen más métodos). Consiste en dividir el nivel de significación entre el número de comparaciones entre grupos. De esta forma, el FWER se mantiene aproximadamente constante.

En el ejemplo anterior en el que se tenían 4 grupos y $\alpha = 0,05$, al aplicar el método de Bonferroni se tendría que $\alpha' = \alpha/n = \alpha/6 = 0,0083$. Por lo tanto, $FWER = 1 - (1 - 0,0083)^6 = 0,0488$. Para que entre 2 de los grupos halla diferencias significativas, su p-valor tendrá que ser menor de 0.0083.

Las comparaciones entre 2 grupos pueden realizarse mediante intervalos de confianza para la diferencia entre medias. Si el 0 está en el intervalo, se podrá afirmar con una confianza $1 - \alpha' = 1 - \alpha/n$ que las medias de ambos grupos son iguales.

A.2. Modelo Lineal de Efectos Mixtos (LME)

Los LME son particularmente interesantes en el análisis de datos longitudinales. Son adecuados para lidiar con problemas típicos en estudios reales:

- La varianza entre sujetos podría no ser constante a lo largo del estudio.
- Pérdida de datos (cabe destacar aquellos sujetos con medidas únicamente en un punto temporal).
- Intervalos temporales no uniformes en las medidas.

Modelan 3 formas de variabilidad en los datos: (1) variación entre sujetos, (2) cambios biológicos en un mismo sujeto, (3) errores en la medida. Matemáticamente, se puede expresar de la siguiente forma:

$$Y_i = X_i\beta + Z_ib_i + e_i$$

Donde:

- i es el identificador de cada sujeto.
- Y_i es un vector de medidas de dimensión $n_i \times 1$.
- n_i es el número de medidas en el tiempo.
- X_i es la matriz $n_i \times p$ de efectos fijos.
- β es un vector $p \times 1$ con los coeficientes de regresión de efectos fijos (desconocido).
- Z_i es la matriz $n_i \times q$, $q \leq p$ de efectos aleatorios. Sus columnas son un subconjunto de X_i , significando que cualquier componente de X_i que se incluya en Z_i puede variar de forma aleatoria.
- b_i es un vector $q \times 1$ de efectos aleatorios. Se asume la distribución: $b_i \sim N(0, D)$.
- e_i es un vector $n_i \times 1$ de errores en la medida. Se asume la distribución: $e_i \sim N(0, \sigma^2 I_{n_i})$.

Se distinguen expresiones para la media condicionada y la media marginal (con la varianza ocurre lo mismo):

- $E(Y_i) = X_i\beta$. Representa la media de la población y es la misma para todos los individuos.
- $E(Y_i|b_i) = X_i\beta + Z_ib_i$. Incluye efectos aleatorios, específicos para cada sujeto. Permite un análisis más individualizado.

El problema se reduce a estimar β , σ y D . Conocidos $\hat{\sigma}$ y \hat{D} , β se estima mediante la fórmula:

$$\hat{\beta} = \left(\sum_{i=1}^m X_i^T \hat{\Sigma}_i^{-1} X_i \right)^{-1} \sum_{i=1}^m X_i^T \hat{\Sigma}_i^{-1} y_i$$

Se puede obtener una estimación insesgada de $\hat{\sigma}$ y \hat{D} maximizando la función (son necesarios métodos numéricos):

$$l_{ReML} = \frac{1}{2} \sum_{i=1}^m \log |\Sigma_i^{-1}| - \frac{1}{2} \sum_{i=1}^m (y_i - X_i \hat{\beta})^T \Sigma_i^{-1} (y_i - X_i \hat{\beta}) - \frac{1}{2} \log \left| \sum_{i=1}^m X_i^T \Sigma_i^{-1} X_i \right|$$

Bernal et al. [25] propone las siguientes expresiones para calcular el tamaño de la muestra requerido para detectar un efecto en particular (A.1) y la potencia de un estudio (A.2).

$$N = \frac{(z_{(1-\alpha/2)} + z_{(1-\gamma)})^2 2\varphi^2}{\delta^2} \tag{A.1}$$

$$1 - \gamma = 1 - F(cv; c, v_e, nc) \tag{A.2}$$

Además, demuestra la capacidad de los LME en el análisis longitudinal:

- Obteniendo resultados concordes a estudios anteriores.
- Comparando con otras técnicas usadas para el mismo fin (p.ej. RM-ANOVA) en términos de especificidad, sensibilidad y repetibilidad.
- Mostrando su desempeño en sets de datos con sujetos con medidas en un único punto temporal.

Apéndice B

Clasificación

B.1. Introducción

Un modelo de clasificación determina la pertenencia de un sujeto a un grupo. Los resultados posibles son varios:

- Verdadero positivo (VP): tanto el valor del modelo como el observado son positivos.
- Verdadero negativo (VN): tanto el valor del modelo como el observado son negativos.
- Falso positivo (FP): el valor del modelo es positivo y el observado negativo.
- Falso negativo (FN): el valor del modelo es negativo y el observado positivo.

Asimismo, conviene definir:

- Sensibilidad (SEN): capacidad del modelo de identificar correctamente los casos positivos: $SEN = VP/(VP + FN)$.
- Especificidad (SPE): capacidad del modelo de identificar correctamente los casos negativos: $SPE = VN/(VN + FP)$.
- Exactitud: proporción de predicciones correctas: $(VN + VP)/(VN + VP + FN + FP)$.
- *Balanced Accuracy* (BCA): promedio entre la sensibilidad y la especificidad: $(SEN + SPE)/2$. Es útil en poblaciones cuyas clases están desbalanceadas. Si la mayoría de los sujetos (p.ej. un 90 %) pertenecen a una clase, es muy probable que el modelo tienda a clasificarlos dentro de dicha clase. Esto podría dar lugar a valores muy elevados en la exactitud y en la sensibilidad o especificidad (según predominen los VP o los VN), perdiendo información de la clase minoritaria. La BCA, por el contrario, proporciona una medida más estable que incorpora información de ambas clases.

La cantidad de VP, VN, FP, FN puede resumirse en una matriz de confusión.

B.2. Curva ROC y AUC

La predicción del modelo se basa en la probabilidad que tiene el sujeto de pertenecer a cada una de las clases. En un problema de clasificación binaria (2 grupos), se establece un umbral de probabilidad. Cuanto más elevado sea el umbral, más difícil será que el modelo clasifique al sujeto como positivo. Un umbral muy exigente puede resultar interesante cuando sea crucial la identificación precisa de los casos positivos (por ejemplo, en

el diagnóstico de pacientes con COVID).

El establecimiento de distintos umbrales da lugar a múltiples matrices de confusión. Interpretar estos resultados puede conducir a confusiones. Por ello, la curva ROC (*Ratio Operation Curve*, figura B.1) resume toda esta información de manera visual. En el eje horizontal se sitúa la tasa de falsos positivos ($1 - SPE$), y en el eje vertical la tasa de verdaderos positivos. La línea recta que pasa por $(0,0)$ y $(1,1)$ es el lugar geométrico de los puntos en los que se cumple $SEN = 1 - SPE$. Los puntos situados por encima de esta línea se traduce en una mayor capacidad de clasificación del modelo. La curva ROC facilita la elección del umbral de probabilidad óptimo.

El área que encierra la curva ROC se denomina área bajo la curva (AUC o *Area Under the Curve*). Cuanto más próximo a 1 esté el AUC, mayor será la capacidad predictiva del modelo. Un $AUC = 0.5$ equivale a una clasificación aleatoria. El AUC es útil a la hora de comparar modelos: el mejor clasificador es aquel cuya AUC sea mayor.

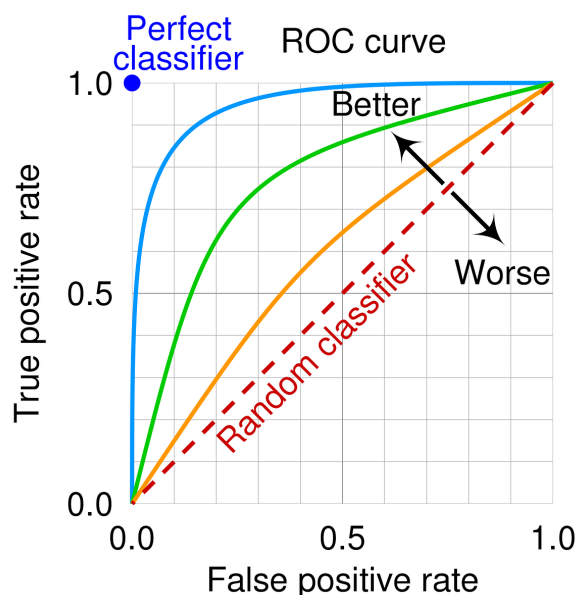


Figura B.1: Curva ROC para distintos clasificadores. La línea recta que une $(0,0)$ con $(1,1)$ representa un clasificador aleatorio. Un clasificador ideal es aquel donde $AUC = 1$.

B.3. Clasificación multiclase

El problema multiclase se aborda transformándolo en uno de clasificación binaria [15]:

- *One-vs-Rest* u OvR: la curva ROC y el AUC se calculan para cada clase, considerando la clase en cuestión como positiva y el resto de clases como negativas.
- *One-vs-One* u OvO: se seleccionan las clases por pares y se calcula la curva ROC y el AUC para cada par posible.

Los resultados globales se calculan por 2 vías:

- *Micro-average*: cada punto de la curva ROC se obtiene SEN y $(1 - SPE)$ considerando los VP, VN, FP, FN para cada caso individual:

$$SEN = \frac{\sum_i VP_i}{\sum_i (VP_i + FN_i)}$$

$$1 - SPE = \frac{\sum_i FP_i}{\sum_i (FP_i + VN_i)}$$

Este enfoque es válido únicamente para OvR.

- *Macro-average*: cada punto de la curva ROC se obtiene SEN y (1 - SPE) como promedio de los valores individuales:

$$SEN = \frac{\sum_i^n SEN_i}{n}$$

$$1 - SPE = \frac{\sum_i^n (1 - SPE_i)}{n}$$

Este enfoque es válido tanto para OvR como OvO.

B.4. Clasificador de Bayes

El clasificador de Bayes es un algoritmo de *machine learning* supervisado, es decir, se entrena con conjuntos de datos etiquetados, usado en problemas de clasificación. Se basa en el Teorema de Bayes:

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

$$= \frac{P(B|A_i)P(A_i)}{\sum_j^n P(B|A_j)P(A_j)}$$

Donde:

- $\{A_1, A_2, \dots, A_n\}$ son los sucesos que pueden ocurrir.
- B es la variable aleatoria que se usa para entrenar el modelo.
- $P(A_i|B)$ es la probabilidad a posteriori. Representa la probabilidad de que ocurra un suceso conocido B .
- $P(B|A_i)$ es la verosimilitud.
- $P(A_i)$ es la probabilidad a priori. Representa la probabilidad de que ocurra un suceso dado el conjunto de datos inicial.
- $P(B)$ es la evidencia.

La probabilidad a priori y la función de densidad de la verosimilitud se hallan durante la fase de entrenamiento. Cuando llega una nueva observación, se calculan la verosimilitud y la evidencia para su valor de B . A continuación, se determina la probabilidad a posteriori para cada posible suceso. Finalmente, se asignará un suceso correspondiente a la nueva observación.

B.4.1. KDE

La estimación de densidad de núcleo (KDE o *Kernel Density Estimation*) es una técnica no paramétrica utilizada en estadística para conocer la función de densidad asociada a un conjunto de datos. En vez de asumir una distribución específica de los datos, típicamente normal, se asigna a cada aparición una función de núcleo que aporta densidad local. El promedio de las densidades locales resulta en la función de densidad de la muestra:

$$\hat{f}(x) = \frac{1}{n} \sum_i^n K_h(x - x_i) = \frac{1}{n \cdot h} \sum_i^n K\left(\frac{x - x_i}{h}\right)$$

$$\int_{-\infty}^{\infty} K_h(x - x_i) dx = \int_{-\infty}^{\infty} \frac{1}{h} K\left(\frac{x - x_i}{h}\right) dx = 1 \iff K_h(x - x_i) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$

Donde:

- n es el número de muestras.
- h es el ancho de banda.
- K_h es la función de núcleo. Puede adoptar diversas formas, por ejemplo: gaussiana, triangular o rectangular.
- $\hat{f}(x)$ es la probabilidad en el punto x.

La elección del ancho de banda determinará la suavidad de la función (figura B.2). El error cuadrático medio integrado (MISE o *Mean Integrated Squared Error*) se usa a menudo para evaluar la calidad de la estimación de la función de densidad. El ancho de banda óptimo será aquel que minimice el MISE:

$$MISE = \mathbb{E} \left(\int (\hat{f}(x) - f(x))^2 dx \right)$$

$$\frac{\partial MISE}{\partial h} = 0$$

Donde:

- \mathbb{E} es la esperanza de la variable aleatoria.
- f(x) es la función de densidad verdadera.

Se han desarrollado numerosos métodos para estimar el ancho de banda óptimo. Cabe nombrar alguna de ellas: regla empírica de Scott, regla empírica de Silverman o validación cruzada.

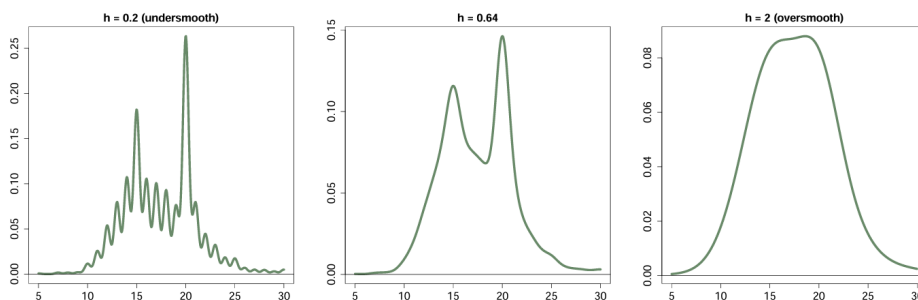


Figura B.2: Funciones de densidad estimadas mediante KDE con diferentes anchos de banda. El valor de h en la figura de la izquierda produce subsuavización (*undersmoothing*). En la figura de la derecha ocurre lo contrario (sobresuavización o *oversmoothing*). En la figura central la elección del parámetro h es la correcta.

Apéndice C

Paralelización de tareas

C.1. Introducción

La ejecución de tareas puede acelerarse aprovechando los procesadores disponibles en el sistema. La ejecución secuencial de las tareas se sustituye por varios procesos (o hilos) que trabajan de zona simultánea. La paralelización es conveniente en tareas de cómputo intensivo que realizadas en un único proceso tendrían tiempos de ejecución muy elevados. Si las tareas necesitan acceder a regiones de memoria compartida, está más indicado crear varios hilos de ejecución. Si están bien diferenciadas y son independientes entre sí es mejor usar procesos, ya que cada proceso dispone de su propio espacio de memoria. En sistemas con varios nodos es más recomendable usar procesos.

En esta sección se discute la paralelización de tareas (en Python y R) usando procesos en distintos sistemas:

- Windows 11 Home
- Ubuntu 20.04.6 LTS (Curie)
- CentOS Linux release 8.3.2011 (Magerit-3)

C.2. Paralelización en R

Se usan los paquetes `foreach` y `doParallel` en Windows. En Linux se usa el paquete `Rmpi` [51].

C.2.1. Paralelización en Windows

Se paralelizan los bucles en los que las iteraciones sean de cómputo intensivo e independientes entre sí, siguiendo los siguientes pasos:

1. Determinar el número de trabajadores paralelos: la función `detectCores` devuelve el número de CPUs en el sistema, que podrán aprovecharse para la paralelización. Conviene dejar al menos un núcleo disponible para otras tareas del sistema, lo que evita sobrecargas. Por lo tanto, al resultado devuelto por `detectCores` se le suele restar 1.
2. Usar la función `makeCluster` para configurar el entorno de trabajo paralelo.
3. Usar la función `registerDoParallel` para que todos los bucles que utilicen la instrucción `foreach` lo hagan según la configuración establecida anteriormente.

4. Realización paralela de los bucles usando `foreach` combinado con `%dopar%`, según la sintaxis:

```
res <- foreach (i = 1:n, .combine=combineFun)%dopar% { #Code }
```

Donde:

- `res` es el resultado devuelto por el bucle después de aplicar la función de combinación sobre las salidas de las iteraciones.
- `%dopar%` especifica que se va a ejecutar el bucle de forma paralela según la configuración establecida.
- `combineFun` es la función de combinación especificada para el argumento `.combine`. Especifica cómo deben combinarse los resultados de cada iteración. Puede ser una función predefinida o una función definida por el usuario. Por ejemplo, el argumento `"rbind"` combina los datos en un *dataframe* por filas, donde cada fila es el *output* de cada iteración.

Es importante destacar que si el bucle utiliza librerías específicas, será necesario volverlas a cargar, aunque hubieran sido cargadas previamente al inicio del *script*. Esto es porque cada proceso tiene su propio entorno de ejecución, distinto del entorno global.

5. Llamar a `stopCluster` para que termine los procesos en ejecución y libere los recursos utilizados.

C.2.2. Paralelización en Linux

Los procesos se crean antes de la ejecución del *script*:

- Curie: con el mandato desde línea de comandos:

```
mpirun -np ntasks Rscript script.R
```

donde `ntasks` define el número de procesos que ejecutarán `script.R`.

- Magerit-3: con un archivo de *shell* `.sh`. El número de trabajadores simultáneos se especifica en las directivas `#SBATCH --nodes=nnodes` y `#SBATCH --ntasks=n`, donde `nnodes` es el número de nodos y `n` es el número total de tareas. La directiva `#SBATCH --tasks-per-node` puede usarse para indicar el número de tareas que se ejecutan en cada nodo.

En el `.sh` deben constar los siguientes mandatos:

1. `module load R/#version` donde la `#version` es la versión de R a utilizar. El mandato `module spider R` devuelve las versiones disponibles de R en el *cluster*.
2. `mkdir -p /carpeta/de/archivos/temporales` para crear una carpeta en la que se guardarán los archivos temporales generados durante la ejecución.
3. `srun --ntasks=$SLURM_NTASKS Rscript /ruta/al/script.R` donde `ntasks` define el número de veces que se lanzará el *script* de R.
4. `rm -rf /carpeta/de/archivos/temporales` para eliminar el contenido de la carpeta de archivos temporales.

En el código C.1 se presenta un ejemplo con el contenido de este archivo.

La paralelización de las tareas se hace usando el estándar MPI. Cuando se ejecuta `mpirun` o `srun`, se crea un grupo de procesos que pueden intercambiar mensajes entre sí por medio de un comunicador. Por defecto, se les asigna el comunicador 0, aunque el paquete `Rmpi` permite crear comunicadores nuevos para grupos de procesos. A continuación, se indican algunas funciones que resultan útiles:

- `mpi.comm.size` devuelve el número de procesos en un comunicador.
- `mpi.comm.rank` devuelve el identificador del proceso dentro de un grupo de procesos.
- `mpi.send.Robj` envía un objeto de R (por ejemplo, un *dataframe*) a otro proceso dentro de un comunicador. Utiliza la función `serialize` para convertir el objeto a una cadena de caracteres binarios.
- `mpi.recv.Robj` recibe un objeto de R desde otro proceso dentro de un comunicador. Usa `unserialize` para regenerar el objeto.
- `mpi.any.source` especifica que un mensaje puede llegar desde cualquier origen. Es de gran utilidad en el argumento `source` de `mpi.recv.Robj`.

C.3. Paralelización en Python

Se usa el paquete `concurrent.futures` en Windows. En Linux se usa el paquete `mpi4py` [26].

C.3.1. Paralelización en Windows

Se paralelizan los bucles en los que las iteraciones sean de cómputo intensivo e independientes entre sí, siguiendo los siguientes pasos:

1. Determinar el número de trabajadores paralelos: la función `os.cpu_count` devuelve el número de CPUs en el sistema.
2. Crear un manejador de contexto que asegure la gestión segura y eficiente de los procesos. Se usa la sintaxis:

```
with ProcessPoolExecutor(max_workers = num_workers - 1) as executor:
```

donde `num_workers - 1` define el número de procesos.

3. Usar la función `map` para aplicar una función a un iterable de manera concurrente. Las iteraciones estarán definidas en una función en vez del cuerpo del bucle. Cada elemento del iterable contiene los argumentos de la función. El resultado de `map` es un elemento iterable con los valores devueltos en cada iteración. La sintaxis se puede resumir de la siguiente forma:

```
for res in executor.map(iterFun, args):
```

Donde:

- `res` es el iterable devuelto por `map`.
- `iterFun` es la función que se ejecuta en cada iteración del bucle.
- `args` es una lista con los argumentos de cada iteración. Cada argumento solo puede contener un único objeto, por lo que es recomendable usar una tupla con todos los argumentos y descomprimirla al inicio de `iterFun`.

C.3.2. Paralelización en Linux

Los procesos se crean antes de la ejecución del *script*:

- Curie: con el mandato desde línea de comandos:

```
mpirun -np ntasks python script.py
```

- Magerit-3: con un archivo de *shell* `.sh`.

En el `.sh` deben constar los siguientes mandatos:

1. `module load Anaconda3/#version`
2. `source ~/.bashrc`
3. `conda activate #entorno` donde `#entorno` es el entorno Conda que se desea cargar.
4. `mkdir -p /carpeta/de/archivos/temporales`
5. `srun --ntasks=$SLURM_NTASKS python /ruta/al/script.py`
6. `rm -rf /carpeta/de/archivos/temporales`

En el código C.2 se presenta un ejemplo con el contenido de este archivo.

La paralelización de las tareas se hace usando el estándar MPI. A continuación, se indican algunas funciones que resultan útiles:

- `MPI.COMM_WORLD` obtiene el comunicador predeterminado.
- `Get_size` devuelve el número de procesos en un comunicador.
- `Get_rank` devuelve el identificador del proceso dentro de un grupo de procesos.
- `send` envía un objeto a otro proceso dentro de un comunicador.
- `recv` recibe un objeto desde otro proceso dentro de un comunicador.
- `MPI.ANY_SOURCE` especifica que un mensaje puede llegar desde cualquier origen.

C.4. Evaluación de la paralelización en Linux

Asegurarse de que la paralelización se haya realizado exitosamente es imprescindible. De lo contrario, el sistema podría estar sobrecargado y tardar más de lo previsto en realizar el trabajo.

La herramienta `htop` sirve para el monitoreo del sistema en tiempo real y resulta de gran utilidad. La figura C.1 muestra un posible resultado de la ejecución de `htop`. Se diferencian 2 zonas:

- Zona superior: proporciona una vista general y resumida del estado del sistema:
 - Uso de recursos: muestra unas barras con el uso de cada CPU disponible en el sistema. Asimismo, unas barras adicionales que indican el uso de memoria RAM y espacio de *swap*.
 - Tareas: indica el número de procesos en el sistema, de hilos, y los procesos que están ejecutándose.

gradualmente hasta que, eventualmente, el sistema operativo lo finaliza. Esta clase de errores pueden ocurrir varias horas después del lanzamiento del programa, por lo que el monitoreo instantáneo con `htop` no es recomendable.

El siguiente mandato permite monitorear el consumo de memoria de forma automática:

```
vmstat -SM intervalo_segundos número_intervalos >
    $SCRATCH/memoryusage_$$SLURM_JOB_ID.out &
```

Este comando registra el uso de memoria en MiB (opción `-SM`), ejecutándose en segundo plano (opción `&`) sin interrumpir el proceso principal. Los parámetros `intervalo_segundos` y `número_intervalos` indican el tiempo que debe pasar entre cada medida y el número de mediciones¹. El resultado se guarda en un fichero vinculado al identificador de la tarea en SLURM. De esta forma, es posible analizar el consumo de memoria posteriormente a la finalización del proceso.

El comando debe incluirse en el fichero `.sh`. Sus parámetros a la salida de `vmstat` se describen seguidamente:

- `procs` (procesos):
 - `r`: número de procesos en ejecución o esperando a ser ejecutados.
 - `b`: número de procesos bloqueados (en espera de la finalización de E/S).
- `memory` (memoria):
 - `swpd`: memoria virtual.
 - `free`: memoria libre disponible.
 - `buff`: memoria usada como *buffer*.
 - `cache`: memoria usada como caché.
- `swap` (memoria de intercambio):
 - `si`: memoria intercambiada desde el disco hacia la RAM.
 - `so`: memoria intercambiada desde la RAM hacia el disco.
- `io`
 - `bi`: bloques por segundo leídos desde un dispositivo de bloque (p.ej. un disco duro).
 - `bo`: bloques por segundo enviados a un dispositivo de bloque.
- `system` (sistema):
 - `in`: número de interrupciones por segundo.
 - `cs`: número de cambios de contexto por segundo.
- `cpu`: son los porcentajes del tiempo total de CPU:
 - `us`: tiempo ejecutando procesos que no son del sistema (tiempod de usuario).
 - `sy`: tiempo ejecutando procesos del sistema.
 - `id`: tiempo inactivo.

¹Por ejemplo, `vmstat -SM 300 576 > $SCRATCH/memoryusage_$$SLURM_JOB_ID.out &` registra el uso de memoria cada 300 segundos (5 minutos) un total de 576 veces. Es decir, monitorea durante 48 horas.

- **wa**: tiempo esperando la finalización de E/S.
- **st**: tiempo robado por máquinas virtuales que corren en el sistema.

La figura C.2 muestra 2 ejemplos de posibles resultados de la ejecución. Mientras que en el primero (figura C.2a) no se produce acumulación de memoria, en el segundo (figura C.2b) la memoria libre disponible desciende con el tiempo.

```
procs -----memory----- --swap-- ----io---- -system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
1 0   585 770370  3   430  0  0  0  0  4  0 85  2 14  0  0
40 0   585 760102  3   677  0  0  5  5 92314 137533 66 11 22  0  0
40 0   585 760100  3   677  0  0  0  0 42129 2092 82  1 17  0  0
40 0   585 760096  3   677  0  0  0  1 42141 2094 82  1 17  0  0
40 0   585 760094  3   677  0  0  0  0 42126 2089 82  1 17  0  0
40 0   585 760092  3   677  0  0  0  0 42120 2091 82  1 17  0  0
40 0   585 760091  3   677  0  0  0  1 42121 2095 82  1 17  0  0
40 0   585 760088  3   677  0  0  0  0 42131 2089 82  1 17  0  0
40 0   585 760087  3   677  0  0  0  0 42113 2090 82  1 17  0  0
40 0   585 760085  3   677  0  0  0  0 42118 2091 82  1 17  0  0
40 0   585 760081  3   677  0  0  0  0 42111 2090 82  1 17  0  0
40 0   585 760080  3   677  0  0  0  0 42107 2089 82  1 17  0  0
40 0   585 760076  3   677  0  0  0  0 42132 2091 82  1 17  0  0
40 0   585 760075  3   677  0  0  0  2 42148 2097 82  1 17  0  0
40 0   585 760075  3   677  0  0  0  0 42114 2090 82  1 17  0  0
```

(a)

```
procs -----memory----- --swap-- ----io---- -system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
8 0   306 769659  3   603  0  0  0  0  1  0 83  3 14  0  0
91 0   306 754335  3   859  0  0  5  2 96361 146461 81 12  6  0
91 0   306 753134  3   859  0  0  0  0 49250 29575 96  2  2  0
107 0   306 751898  3   859  0  0  0  0 49223 29433 96  2  2  0
79 0   306 750748  3   859  0  0  0  0 49290 29238 96  2  2  0
90 0   306 750671  3   859  0  0  0  0 49312 28584 96  2  2  0
74 0   306 750034  3   859  0  0  0  0 49170 28509 96  2  2  0
78 0   306 749228  3   859  0  0  0  0 49277 29001 96  2  2  0
83 0   306 748043  3   859  0  0  0  0 49388 29091 96  2  2  0
86 0   306 747766  3   859  0  0  0  0 49435 27346 96  2  2  0
94 0   306 747226  3   859  0  0  0 12 49240 28010 96  2  2  0
94 0   306 746559  3   859  0  0  0  0 49151 28863 96  2  2  0
95 0   306 745622  3   859  0  0  0  1 49432 28924 96  2  2  0
91 0   306 745356  3   860  0  0  0  1 49668 26893 96  2  2  0
96 0   306 744830  3   860  0  0  0  0 49427 27690 96  2  2  0
```

(b)

Figura C.2: Posible resultado de la ejecución de `vmstat -SM intervalo_segundos número_intervalos > $SCRATCH/memoryusage.$SLURM_JOB_ID.out &`. La figura de la izquierda muestra un trabajo en el que no se produce acumulación de memoria, mientras que en la derecha la memoria libre desciende con el tiempo.

C.6. Aceleración de tareas con CUDA/GPU

Las GPU (*Graphic Processing Unit*) son circuitos electrónicos diseñados para acelerar el procesamiento y visualización de gráficos. A diferencia de las CPU (*Central Processing Unit*), adecuadas para el procesamiento en serie, las GPU son capaces de realizar muchas operaciones a la vez y son ideales para el procesamiento paralelo. Las primeras presentan una baja latencia² y disponen de unos pocos núcleos. La latencia de las segundas es más alta, pero presentan una mayor cantidad de núcleos.

²La latencia es el tiempo que tarda el procesador en dar respuesta a una instrucción.

CUDA (*Compute Unified Device Architecture*) es una plataforma de desarrollo de *software* utilizada para acelerar el cálculo paralelo mediante programas que se ejecutan en dispositivos compatibles con CUDA. Es compatible con la mayoría de sistemas operativos y permite dividir una tarea en miles de hilos que corren de forma concurrente. Es una tecnología desarrollada por NVIDIA que surgió a mediados de los 2000 y originalmente fue usada en el procesamiento de gráficos. Actualmente, se ha extendido a otras industrias como la minería de datos, aprendizaje automático o finanzas computacionales.

C.6.1. Aceleración en Magerit-3

Basta con asegurarse de que la aplicación esté adaptada para hacer uso de CUDA e incluir las siguientes directivas SLURM:

- `#SBATCH --partition=standard-gpu`: solicita una partición que contiene GPUs.
- `#SBATCH --gres=gpu:<type>:<# gpus>`: indica el tipo y el número de GPUs requeridas. El tipo puede ser:
 - `a100`: una tarjeta NVIDIA A100 [5].
 - `v100`: una tarjeta NVIDIA V100 [10].

En el caso de no tener ninguna preferencia de modelo de GPU, es posible omitir el tipo (`#SBATCH --gres=gpu:<# gpus>`), de forma que se asignará la primera disponible.

```
1 #!/bin/bash
2 ##----- Start job description -----
3 #SBATCH --partition=standard
4 #SBATCH --job-name=multi_Grace
5 #SBATCH --nodes=1
6 #SBATCH --ntasks=40
7 #SBATCH --mem-per-cpu=4096
8 #SBATCH --time=50:00:00
9 #SBATCH --mail-type=ALL
10 #SBATCH --mail-user=@mail_address
11 #SBATCH --output=out-%j.log
12 #SBATCH --error=err-%j.log
13 #SBATCH --chdir=/home/v597/v597521/multigrace/
14 ##----- End job description -----
15
16 #Load R version
17 module load R/4.2.2-foss-2022b
18
19 #Create scratch directory for temporary files
20 export SCRATCH="/home/v597/v597521/multigrace"
21 mkdir -p $SCRATCH/$SLURM_JOB_ID
22
23 #Run job
24 srun --ntasks=$SLURM_NTASKS Rscript /home/v597/v597521/multigrace/
    multigrace_par_mpi.R
25
26 #Remove files from scratch directory
27 rm -rf $SCRATCH/$SLURM_JOB_ID
```

Código C.1: Código .sh para lanzar un trabajo en R en Magerit-3. Se solicitan un total de 40 tareas distribuidas en 1 único nodo. El tiempo máximo de ejecución del trabajo es de 50 horas.

```
1 #!/bin/bash
2 ##----- Start job description -----
3 #SBATCH --partition=standard
4 #SBATCH --job-name=multi_Leaspy
5 #SBATCH --nodes=1
6 #SBATCH --ntasks=40
7 #SBATCH --mem-per-cpu=4096
8 #SBATCH --time=50:00:00
9 #SBATCH --mail-type=ALL
10 #SBATCH --mail-user=@mail_address
11 #SBATCH --output=out-%j.log
12 #SBATCH --error=err-%j.log
13 #SBATCH --chdir=/home/v597/v597521/multileaspy/
14 ##----- End job description -----
15
16 #Load conda environment
17 module load Anaconda3/2024.02-1
18 source ~/.bashrc
19 conda activate leaspy
20
21 #Create scratch directory for temporary files
22 export SCRATCH="/home/v597/v597521/multileaspy"
23 mkdir -p $SCRATCH/$SLURM_JOB_ID
24
25 #Run job
26 srun --ntasks=$SLURM_NTASKS python /home/v597/v597521/multileaspy/
    multileaspy_par_mpi.py
27
28 #Remove files from scratch directory
29 rm -rf $SCRATCH/$SLURM_JOB_ID
```

Código C.2: Código .sh para lanzar un trabajo en python en Magerit-3. Se solicitan un total de 40 tareas distribuidas en 1 único nodo. El tiempo máximo de ejecución del trabajo es de 50 horas.

Apéndice D

Originalidad del documento

Turnitin [22] es una herramienta especializada en la detección de plagio ampliamente utilizada en el ámbito académico para verificar la originalidad de los documentos. Asimismo, ayuda a prevenir el contenido generado con sistemas de inteligencia artificial como ChatGPT. El texto recibido se coteja con una base de datos que incluye artículos académicos, contenido de Internet o trabajos previos enviados por otros estudiantes. El análisis revela posibles coincidencias con otras fuentes, así como el porcentaje de similitud.

La figura D.1 muestra un resumen del informe de Turnitin sobre este documento.

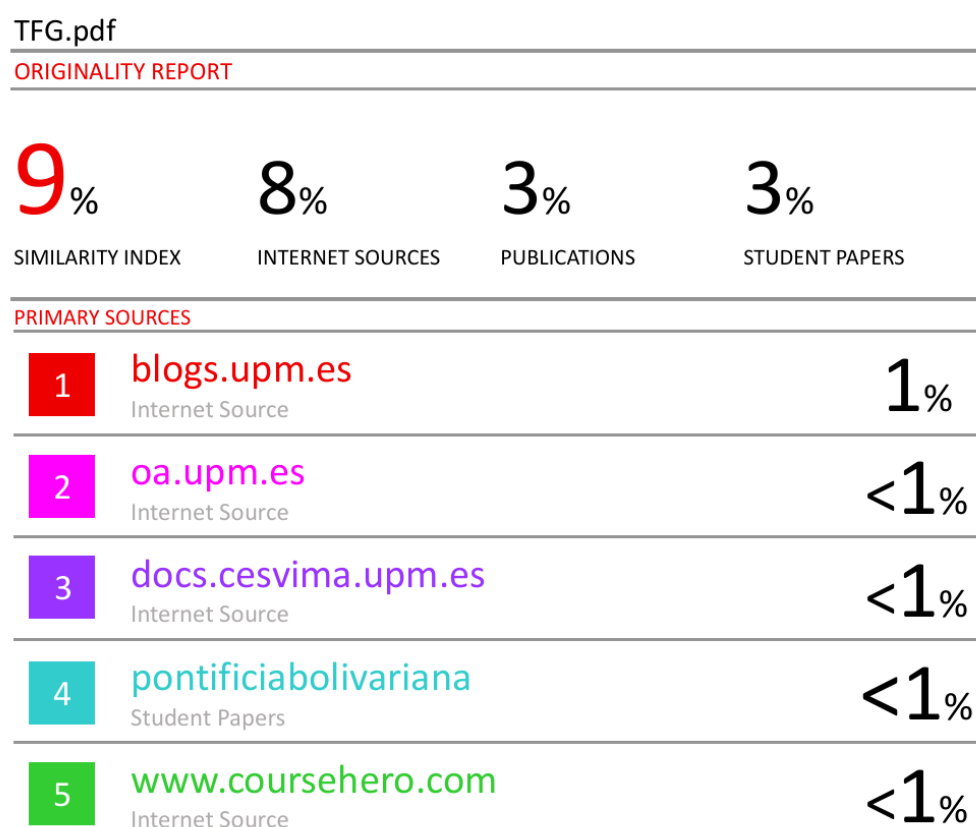


Figura D.1: Resumen del informe de similitud de Turnitin.

Según se observa, el índice de similaridad es del 10%. El informe completo puede encontrarse en `/demo/originalidad/TFG_turnitin.pdf`.

Lista de Acrónimos

APOE ϵ 4 Alelo ϵ 4 de la Apolipoproteína E. 14, 15, 39, 49, 83

A β Beta amiloide. 2–4, 9, 14, 15, 23, 48, 49, 83, 132, 138, 142

A β ₄₂ Péptido 42 de beta amiloide. 9

ACC *Accuracy* (Precisión). 26

AD *Alzheimer's Disease* (Enfermedad de Alzheimer). 2, 9, 10, 12–14, 24, 26, 28, 29, 37, 44, 46, 48, 49, 60, 83, 126, 138

Adam *Adaptive Moment Estimation* (Estimación Adaptativa de Momentos). 29

ADAS-Cog *Alzheimer's Disease Assessment Scale - Cognitive Subscale* (Escala de Evaluación Cognitiva para la Enfermedad de Alzheimer). 42, 46, 138

ADAS11 *Alzheimer's Disease Assessment Scale - Cognitive Subscale (11 items)* (Escala de Evaluación de la Enfermedad de Alzheimer – Subescala Cognitiva (11 ítems)). 37, 46

ADAS13 *Alzheimer's Disease Assessment Scale - Cognitive Subscale (13 items)* (Escala de Evaluación de la Enfermedad de Alzheimer – Subescala Cognitiva (13 ítems)). 14, 25–27, 29, 30, 37, 46, 61, 69, 78, 83, 91, 109, 124, 125, 129, 132, 138, 139

ADASQ4 *ADAS-Cog Question 4* (Pregunta 4 de la Subescala Cognitiva de ADAS). 46

ADCS-PACC *Alzheimer's Disease Cooperative Study - Preclinical Alzheimer Cognitive Composite* (Estudio Cooperativo sobre la Enfermedad de Alzheimer - Compuesto Cognitivo Preclínico del Alzheimer). 47

ADN Ácido Desoxirribonucleico. 1

ADNI *Alzheimer's Disease Neuroimaging Initiative* (Iniciativa para la Neuroimagen de la Enfermedad de Alzheimer). 13–15, 23, 24, 27, 42, 46, 47, 49, 57, 69, 71, 82, 138, 145

ADNIMERGE Base de datos de ADNI. 57–60, 64, 67, 73

ANOVA *Analysis of Variance* (Análisis de Varianza). 60, 82

ApoE Apolipoproteína E. 2

ATP *Adenosine Triphosphate* (Adenosín Trifosfato). 1

AUC *Area Under the Curve* (Área Bajo la Curva). 23, 26, 29, 30, 70–73, 77, 120–123, 153

AV-1451 *Florbetapir* (Trazador PET para amiloide). 49

- BCA** *Balanced Classification Accuracy* (Precisión de Clasificación Balanceada). 77, 121–123, 152
- BCE** *Binary Cross Entropy* (Entropía Cruzada Binaria). 66, 68, 93
- BIC** *Bayesian Information Criterion* (Criterio de Información Bayesiano). 23
- CAR** *Continuous-time AutoRegressive* (Autorregresivo en Tiempo Continuo). 30, 31, 33, 34
- CAR-GRU** *Continuous-time AutoRegressive GRU* (Unidad Recurrente con Puertas Autorregresiva en Tiempo Continuo). 31, 32, 34
- CAR-LSTM** *Continuous-time AutoRegressive Long Short-Term Memory* (Memoria Larga-Corta Autorregresiva en Tiempo Continuo). 31, 32, 34
- CAR-RNN** *Continuous-time AutoRegressive Recurrent Neural Network* (Red Neuronal Recurrente Autorregresiva en Tiempo Continuo). 31, 32, 34
- CARRNN** *Continuous-time AutoRegressive RNN* (Red Neuronal Recurrente Autorregresiva en Tiempo Continuo). 31–33, 45
- CDR** *Clinical Dementia Rating* (Evaluación Clínica de la Demencia). 47
- CDR-Global** *Clinical Dementia Rating - Global* (Valoración Clínica de Demencia - Global). 42
- CDRSB** *Clinical Dementia Rating - Sum of Boxes* (Escala Clínica de Demencia - Suma de Ítems). 25–27, 42, 69, 91–93, 109, 112, 120, 124, 125, 132, 138, 139, 142
- cFSGL** *Convex Fused Sparse Group Lasso* (Lasso de Grupo Escaso Fusionado Convexo). 43
- CN** *Cognitively Normal* (Cognitivamente Normal). 14, 24, 37, 61, 66, 67, 70, 72, 138
- CNN** *Convolutional Neural Network* (Red Neuronal Convolutacional). 34
- CNS** *Central Nervous System* (Sistema Nervioso Central). 47
- COVID** *Coronavirus Disease* (Enfermedad del Coronavirus). 153
- CPU** *Central Processing Unit* (Unidad Central de Procesamiento). 53, 156, 158–162
- CSF** *Cerebrospinal Fluid* (Líquido Cefalorraquídeo). 9, 12, 14, 23, 24, 26, 46, 47, 60, 77, 82, 83, 89, 138, 142
- CU** *Cognitively Unimpaired* (Sin Deterioro Cognitivo). 10–12, 24–26, 28, 59, 60, 64, 75–77, 82, 83, 120, 122, 123, 126, 138–140, 142
- CUDA** *Compute Unified Device Architecture* (Arquitectura Unificada de Cómputo en Dispositivos). 163
- DIGITSCOR** *Digit Symbol Substitution Test Score* (Puntuación en el Test de Sustitución de Símbolos). 47
- DPM** *Disease Progression Model* (Modelo de Progresión de la Enfermedad). 24–27, 34, 38, 44, 60, 73–75, 77, 79, 139, 143

- DPS** *Disease Progression Score* (Puntuación de Progresión de la Enfermedad). 20–22, 24, 65, 70–72, 74, 77, 79, 109, 140
- DSST** *Digit Symbol Substitution Test* (Test de Sustitución de Símbolos). 47
- E-M** *Expectation-Maximization* (Algoritmo Esperanza-Maximización). 18
- Ecog** *Everyday Cognition* (Cognición Cotidiana). 47
- ECT** *Entorhinal Cortex Thickness* (Espesor de la Corteza Entorrinal). 48
- EMCI** *Early Mild Cognitive Impairment* (Deterioro Cognitivo Leve Temprano). 14, 126, 142
- FAQ** *Functional Activities Questionnaire* (Cuestionario de Actividades Funcionales). 26, 27, 47, 69, 91–93, 132, 138, 139
- FDG** *Fluorodeoxyglucose* (Fluorodesoxiglucosa). 49
- FF** *Forward Filling* (Imputación hacia adelante). 27, 30, 43
- FN** Falso Negativo. 152, 153
- FP** Falso Positivo. 152, 153
- FTP** *File Transfer Protocol* (Protocolo de Transferencia de Archivos). 50
- FTPS** *File Transfer Protocol Secure* (Protocolo de Transferencia de Archivos Seguro). 50
- FWER** *Family-Wise Error Rate* (Tasa de Error Familiar). 149
- GDS** *General Deterioration Scale* (Escala de Deterioro General). 3, 4
- GNU** *GNU's Not Unix* (GNU No es Unix). 50
- GPU** *Graphics Processing Unit* (Unidad de Procesamiento Gráfico). 52, 162, 163
- GRU** *Gated Recurrent Unit* (Unidad Recurrente con Puertas). 34, 43
- HORD** *Hyperparameter Optimization via RBF and Dynamic Coordinate Search* (Optimización de Hiperparámetros mediante RBF y Búsqueda Dinámica de Coordenadas). 29
- ICA** *Independent Component Analysis* (Análisis de Componentes Independientes). 18
- ICV** *Intracranial Volume* (Volumen intracraneal). 37
- IDE** *Integrated Development Environment* (Entorno de Desarrollo Integrado). 50, 56
- INE** Instituto Nacional de Estadística. 5
- IPT** *Inferior Parietal Thickness* (Espesor del Lóbulo Parietal Inferior). 48
- ITT** *Inferior Temporal Thickness* (Espesor del Lóbulo Temporal Inferior). 48
- KDE** *Kernel Density Estimation* (Estimación de Densidad por Núcleo). 20, 22, 70, 154, 155

- L-BFGS** *Limited-memory Broyden-Fletcher-Goldfarb-Shanno* (Algoritmo BFGS de Memoria Limitada). 20
- Lasso** *Least Absolute Shrinkage and Selection Operator* (Operador de Reducción y Selección Absoluta Mínima). 43
- LCR** Líquido Cefalorraquídeo. 3
- LDA** *Linear Discriminant Analysis* (Análisis discriminante lineal). 38
- LDELTOTAL** *Logical Memory Delayed Recall Total* (Memoria Lógica - Recuerdo Diferido Total). 25, 26, 47, 69, 132, 138, 139
- LF** *Linear Filling* (Imputación lineal). 27, 30, 43
- LM** *Logical Memory* (Memoria Lógica). 47
- LMCI** *Late Mild Cognitive Impairment* (Deterioro Cognitivo Leve Tardío). 14, 126, 142
- LME** *Linear Mixed Effects* (Efectos Mixtos Lineales). 24, 63, 150, 151
- LSS** *Long Short Sequence* (Secuencia Larga-Corta). 27, 29
- LSS-MF** *Long Short Sequence with Model Filling* (Secuencia Larga-Corta con Imputación por Modelo). 29, 30
- LSTM** *Long Short Term Memory* (Memoria a Largo y Corto Plazo). 7, 27, 29–31, 34, 35, 40–42, 45, 68, 70, 92, 93, 111, 112, 121, 123–128, 132, 136, 139, 140, 142
- LSTM-FF** *Long Short-Term Memory with Forward Filling* (Memoria a Corto y Largo Plazo con Imputación por hacia adelante). 37
- LSTM-M** *Long Short-Term Memory with Mean Imputation* (Memoria a Corto y Largo Plazo con Imputación por la Media). 37
- LSTM-MF** *Long Short-Term Memory with Model Filling* (Memoria a Corto y Largo Plazo con Imputación por Modelo). 29, 30
- LSTM-T** *Temporal Long Short-Term Memory* (Memoria a Corto y Largo Plazo Temporal). 43
- MAE** *Mean Absolute Error* (Error Absoluto Medio). 23, 29, 30, 37, 38, 42, 43, 64, 66, 68, 70, 71, 73, 75, 77, 90–93, 112, 128, 139–142
- mAUC** *Mean Area Under the Curve* (Área bajo la curva media). 37, 38
- MCI** *Mild Cognitive Impairment* (Deterioro Cognitivo Leve). 10, 11, 24–26, 28, 37, 59, 60, 66, 70, 72, 74–77, 82, 83, 120, 122, 123, 138, 139, 141, 143
- MCMC-SAEM** *Markov Chain Monte Carlo Stochastic Approximation Expectation-Maximization* (Cadena de Markov Monte Carlo con Algoritmo SAEM). 18, 20
- MeanF** *Mean Filling* (Imputación por media). 43
- MeT** *Mean Thickness* (Suma de los Espesores). 49
- MF** *Model Filling* (Imputación por modelo). 27, 30, 43
- MiB** *Mebibyte*. 161

- MISE** *Mean Integrated Squared Error* (Error Cuadrático Integrado Medio). 155
- MMSE** *Mini-Mental State Examination* (Examen Cognitivo Mini-Mental). 27, 37, 42, 46, 69, 83, 91, 92, 109, 112, 124, 132, 139, 142
- MoCA** *Montreal Cognitive Assessment* (Evaluación Cognitiva de Montreal). 47
- mPACCdigit** *Modified Preclinical Alzheimer Cognitive Composite Digit* (Composito Cognitivo Preclínico Modificado Dígitos). 69, 90, 91, 93
- mPACCtrailsB** *Modified Preclinical Alzheimer Cognitive Composite Trails B* (Composito Cognitivo Preclínico Modificado Trails B). 69, 90–93
- MPI** *Message Passing Interface* (Interfaz de Paso de Mensajes). 57, 70, 79, 158, 159
- MRE** *Mean Relative Error* (Error relativo medio). 37, 38
- MRI** *Magnetic Resonance Imaging* (Imagen por Resonancia Magnética). 9, 14, 23, 34, 35, 37–39, 42
- mRMR** *Minimum Redundancy Maximum Relevance* (Mínima Redundancia Máxima Relevancia). 24
- MRNN** *Multi-directional Recurrent Neural Network* (Red neuronal recurrente multidireccional). 37
- MTT** *Medial Temporal Thickness* (Espesor del Lóbulo Temporal Medial). 48
- N-MNIST** *Neuromorphic MNIST* (MNIST Neuronal). 33
- NACC** *National Alzheimer's Coordinating Center* (Centro Nacional de Coordinación del Alzheimer). 23
- NaN** *Not a Number* (No es un Número). 61, 64, 72, 73, 111–113
- NAV** *Normalized Amygdala Volume* (Volumen de la Amígdala Cerebral Normalizado). 48
- NCV** *Normalized Caudate Volume* (Volumen del Caudete Normalizado). 48
- NEV** *Normalized Entorhinal Cortex Volume* (Volumen de la Corteza Entorrinal Normalizado). 48
- NFV** *Normalized Fusiform Gyrus Volume* (Volumen del Giro Fusiforme Normalizado). 48
- NHV** *Normalized Hippocampal Volume* (Volumen Hipocampal Normalizado). 48, 132, 142
- NINDSS** *National Institute of Neurological Disorder and Stroke Study* (Instituto Nacional de Trastornos Neurológicos y Accidentes Cerebrovasculares). 1
- NM** *Neuropsychological Measures* (Medidas Neuropsicológicas). 25, 26, 46, 60, 77, 82, 83, 89, 132
- NMV** *Normalized Medial Temporal Lobe Volume* (Volumen del Lóbulo Temporal Medial Normalizado). 48
- NPav** *Normalized Pallidum Volume* (Volumen del Globo Pálido Normalizado). 48

- NPV** *Normalized Putamen Volume* (Volumen del Putamen Normalizado). 48
- NVEN** *Normalized Lateral Ventricle Volume* (Volumen del Ventrículo Lateral Normalizado). 48
- NVIDIA** *NVIDIA Corporation* (Empresa NVIDIA). 163
- NVV** *Normalized Ventricular Volume* (Volumen Ventricular Normalizado). 48
- OMS** Organización Mundial de la Salud. 4
- OvO** *One-vs-One* (Uno contra Uno). 153, 154
- OvR** *One-vs-Rest* (Uno contra el Resto). 153, 154
- PACC** *Preclinical Alzheimer Cognitive Composite* (Composite Cognitivo Preclínico del Alzheimer). 25–27, 139
- PAQUID** *Personnes Agées Quid* (Estudio PAQUID de personas mayores en Francia). 13
- PCT** *Posterior Cingulate Thickness* (Espesor de la Corteza Cingulada Posterior). 48
- pCU** *Progressive Cognitively Unimpaired* (Sin Deterioro Cognitivo Progresivo). 24–26, 29, 58, 59, 66, 70, 72–77, 82, 83, 89–93, 109–112, 120–128, 132, 138–142
- PET** *Positron Emission Tomography* (Tomografía por Emisión de Positrones). 9, 12, 23, 38, 39, 42, 46, 49, 77
- PET-FDG** *Positron Emission Tomography using Fluorodeoxyglucose* (Tomografía por Emisión de Positrones mediante Fluorodesoxiglucosa). 3, 9
- PiB PET** *Pittsburgh Compound B Positron Emission Tomography* (Tomografía por Emisión de Positrones con Compuesto B de Pittsburgh). 14
- PLSTM** *Peephole Long Short-Term Memory* (Memoria a largo plazo con compuertas peephole). 34, 43
- PLSTM-Z** *Peephole Long Short-Term Memory with Zero Imputation* (LSTM peephole con imputación cero). 37
- pMCI** *Progressive Mild Cognitive Impairment* (Deterioro Cognitivo Leve Progresivo). 29, 75, 76
- PRT** *Precuneus Thickness* (Espesor del Precúneo). 49
- pSMC** Queja Subjetiva Progresiva. 58, 59, 82, 89, 109, 110, 138, 140
- pTau** Proteína Tau fosforilada. 9, 14, 48, 83
- RAM** *Random Access Memory* (Memoria de Acceso Aleatorio). 52, 53, 80, 159–161
- RAVLT** *Rey Auditory Verbal Learning Test* (Prueba de Aprendizaje Verbal Auditivo de Rey). 23, 47, 138, 139
- RAVLT Forgetting** *Rey Auditory Verbal Learning Test Forgetting* (Olvido en la Prueba de Aprendizaje Verbal Auditivo de Rey). 66, 91

- RAVLT Immediate** *Rey Auditory Verbal Learning Test - Immediate Recall* (RAVLT - Recuerdo Inmediato). 27, 69, 93
- RAVLT Learning** *Rey Auditory Verbal Learning Test - Learning Score* (RAVLT – Puntuación de Aprendizaje). 27, 61, 69, 93, 132, 138, 139
- RAVLT Perc Forgetting** Porcentaje de Olvido en la Prueba de Aprendizaje Verbal Auditivo de Rey. 69, 93
- RAVLT-IR** *Rey Auditory Verbal Learning Test - Immediate Recall* (Prueba de Aprendizaje Verbal Auditivo de Rey - Recuerdo Inmediato). 23
- RBF** *Radial Basis Function* (Función de Base Radial). 43
- RM-ANOVA** *Repeated Measures Analysis of Variance* (Análisis de Varianza de Medidas Repetidas). 151
- rMCI** *Reverter Mild Cognitive Impairment* (Deterioro Cognitivo Leve Reversor). 29, 75, 76
- RMSE** *Root Mean Square Error* (Raíz del error cuadrático medio). 37
- RNN** *Recurrent Neural Network* (Red Neuronal Recurrente). 27, 30–32, 34, 39, 40, 121–123, 125
- RNN-MF** *Recurrent Neural Network with Model Filling* (Red Neuronal Recurrente con Imputación por Modelo). 29, 30
- ROC** *Receiver Operating Characteristic* (Curva Característica Operativa del Receptor). 153, 154
- RSS** *Residual Sum of Squares* (Suma de los Cuadrados Residuales). 14
- SAEM** *Stochastic Approximation Expectation-Maximization* (Algoritmo de Esperanza-Maximización con Aproximación Estocástica). 18
- sCU** *Stable Cognitively Unimpaired* (Sin Deterioro Cognitivo Estable). 24–26, 29, 58, 59, 66, 70, 73, 75–77, 82, 83, 89, 109, 120–123, 125–128, 132, 138, 139, 141, 142
- SEMOR** *Self-Modelling Regression* (Regresión Autodefinida). 13
- SEN** *Sensitivity* (Sensibilidad). 26, 152–154
- SEN** Sociedad Española de Neurología. 5
- SFTP** *Secure File Transfer Protocol* (Protocolo Seguro de Transferencia de Archivos). 50
- SLURM** *Simple Linux Utility for Resource Management* (Utilidad Simple de Linux para la Gestión de Recursos). 53, 80, 161, 163
- SMC** Queja Subjetiva de Memoria. 60, 61, 64, 82, 83, 89, 126, 138, 142
- sMCI** *Stable Mild Cognitive Impairment* (Deterioro Cognitivo Leve Estable). 29, 58, 64, 66, 70, 72, 73, 75–77, 83, 109, 120–123, 125, 126, 128, 132, 139–142
- SPE** *Specificity* (Especificidad). 26, 152–154
- SPT** *Superior Parietal Thickness* (Espesor del Lóbulo Parietal Superior). 48

- SSH** *Secure Shell* (Shell Seguro). 49, 51
- sSMC** Queja Subjetiva Estable. 58, 82, 89, 109, 138
- SVM** *Support Vector Machine* (Máquina de Vectores de Soporte). 27, 29, 30
- SVR** *Support Vector Regression* (Regresión de Vectores de Soporte). 27, 29, 43
- T1-MRI** *T1-weighted Magnetic Resonance Imaging* (Imagen por Resonancia Magnética ponderada en T1). 24, 46, 77, 142
- TADPOLE** *The Alzheimer's Disease Prediction Of Longitudinal Evolution* (Predicción de la evolución longitudinal del Alzheimer). 27, 37
- tau Pet** *Tau Positron Emission Tomography* (Tomografía por Emisión de Positrones con trazadores de Tau). 9
- TCP** *Transmission Control Protocol* (Protocolo de Control de Transmisión). 49
- TPT** *Temporal Pole Thickness* (Espesor del Polo Temporal). 49
- TRABSCOR** *Trail Making Test Score* (Puntuación en el Test de Trazado). 47, 69, 93, 139
- tTau** Proteína Tau total. 9
- UCI** *University of California, Irvine* (repositorio de conjuntos de datos públicos). 33
- VN** Verdadero Negativo. 152, 153
- VP** Verdadero Positivo. 152, 153

Bibliografía

- [1] Adni. <https://ida.loni.usc.edu/home/projectPage.jsp?project=ADNI>.
- [2] Anaconda. <https://www.anaconda.com/>.
- [3] Cevima. <https://www.cesvima.upm.es/>.
- [4] Filezilla. <https://filezilla-project.org/>.
- [5] Gpu nvidia a100 con tensor core. <https://www.nvidia.com/es-es/data-center/a100/>.
- [6] How to remove stale models from gpu memory. <https://github.com/keras-team/keras/issues/5345>.
- [7] Latex. <https://www.latex-project.org/>.
- [8] Matlab. <https://es.mathworks.com/products/matlab.html>.
- [9] memory leak when using tensorflow. <https://github.com/keras-team/keras/issues/2102>.
- [10] memory leak when using tensorflow. <https://www.nvidia.com/es-es/data-center/>.
- [11] Miktex. <https://miktex.org/>.
- [12] Putty. <https://www.putty.org/>.
- [13] Python. <https://www.python.org/>.
- [14] R. https://www.r-project.org/?utm_source=chatgpt.com.
- [15] Receiver operating characteristic (roc). https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html.
- [16] Rstudio ide. <https://posit.co/downloads/>.
- [17] Slurm. <https://docs.cesvima.upm.es/magerit/jobs/>.
- [18] Spyder. <https://docs.spyder-ide.org/current/panes/ipythonconsole.html>.
- [19] Strawberry perl. <https://strawberryperl.com/>.
- [20] Texstudio. <https://www.texstudio.org/>.
- [21] tmux. <https://github.com/tmux/tmux/wiki>.
- [22] turnitin. <https://www.turnitin.com/>.
- [23] Xming x server. <http://www.straightrunning.com/XmingNotes/>.

- [24] Petronilla Battista, Christian Salvatore, and Isabella Castiglioni. Optimizing neuropsychological assessments for cognitive, behavioral, and functional impairment classification: a machine learning study. *Behavioural neurology*, 2017(1):1850909, 2017.
- [25] Jorge L Bernal-Rusiel, Douglas N Greve, Martin Reuter, Bruce Fischl, Mert R Sabuncu, Alzheimer’s Disease Neuroimaging Initiative, et al. Statistical analysis of longitudinal neuroimage data with linear mixed effects models. *Neuroimage*, 66:249–260, 2013.
- [26] Lisandro Dalcin et al. *mpi4py: MPI for Python*. mpi4py, 2023. Version 3.1.4.
- [27] Silvia De Francesco, Samantha Galluzzi, Nicola Vanacore, Cristina Festari, Paolo Maria Rossini, Stefano F Cappa, Giovanni B Frisoni, and Alberto Redolfi. Norms for automatic estimation of hippocampal atrophy and a step forward for applicability to the italian population. *Frontiers in Neuroscience*, 15:656808, 2021.
- [28] Michael C Donohue, Hélène Jacqmin-Gadda, Mélanie Le Goff, Ronald G Thomas, Rema Raman, Anthony C Gamst, Laurel A Beckett, Clifford R Jack Jr, Michael W Weiner, Jean-François Dartigues, et al. Estimating long-term multivariate progression from short-term data. *Alzheimer’s & Dementia*, 10:S400–S410, 2014.
- [29] S Duke Han, Caroline P Nguyen, Nikki H Stricker, and Daniel A Nation. Detectable neuropsychological differences in early preclinical alzheimer’s disease: A meta-analysis. *Neuropsychology review*, 27:305–325, 2017.
- [30] EpData. Cifras del alzheimer en españa: número de personas, mortalidad y muertes, s.f.
- [31] Mostafa Mehdipour Ghazi, Mads Nielsen, Akshay Pai, Marc Modat, M Jorge Cardoso, Sébastien Ourselin, and Lauge Sørensen. Robust parametric modeling of alzheimer’s disease progression. *Neuroimage*, 225:117460, 2021.
- [32] Mostafa Mehdipour Ghazi, Lauge Sørensen, Sébastien Ourselin, and Mads Nielsen. Carrnn: A continuous autoregressive recurrent neural network for deep representation learning from sporadic temporal data. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):792–802, 2022.
- [33] Pedro Gil Gregorio. *Neurodegeneración: Alzheimer, Parkinson y ELA*. Neurociencia y psicología. Emse Edapp, S.L., 2018.
- [34] Oskar Hansson, John Seibyl, Erik Stomrud, Henrik Zetterberg, John Q Trojanowski, Tobias Bittner, Valeria Lifke, Veronika Corradini, Udo Eichenlaub, Richard Batrla, et al. Csf biomarkers of alzheimer’s disease concord with amyloid- β pet and predict clinical progression: a study of fully automated immunoassays in biofinder and adni cohorts. *Alzheimer’s & dementia*, 14(11):1470–1481, 2018.
- [35] Clifford R Jack Jr, David A Bennett, Kaj Blennow, Maria C Carrillo, Billy Dunn, Samantha Budd Haeberlein, David M Holtzman, William Jagust, Frank Jessen, Jason Karlawish, et al. Nia-aa research framework: toward a biological definition of alzheimer’s disease. *Alzheimer’s & Dementia*, 14(4):535–562, 2018.
- [36] Wonsik Jung, Eunji Jun, Heung-Il Suk, Alzheimer’s Disease Neuroimaging Initiative, et al. Deep recurrent model for individualized prediction of alzheimer’s disease progression. *NeuroImage*, 237:118143, 2021.

- [37] Young Ju Kim, Si Eun Kim, Alice Hahn, Hyemin Jang, Jun Pyo Kim, Hee Jin Kim, Duk L Na, Juhee Chin, and Sang Won Seo. Classification and prediction of cognitive trajectories of cognitively unimpaired individuals. *Frontiers in Aging Neuroscience*, 15:1122927, 2023.
- [38] Igor Koval. Learning multimodal digital models of disease progression from longitudinal data: methods & algorithms for the description, prediction and simulation of Alzheimer's disease progression. PhD thesis, Institut polytechnique de Paris, 2020.
- [39] Minh Nguyen, Tong He, Lijun An, Daniel C Alexander, Jiashi Feng, BT Thomas Yeo, Alzheimer's Disease Neuroimaging Initiative, et al. Predicting alzheimer's disease progression using deep recurrent neural networks. *NeuroImage*, 222:117203, 2020.
- [40] Oficina de Ciencia y Tecnología del Congreso de los Diputados (Oficina C). Avances en neurociencia: aplicaciones e implicaciones éticas, 2023. Informe C.
- [41] Organización Mundial de la Salud. La cantidad de personas con demencia se triplicará en los próximos 30 años, 2017.
- [42] Sid E O'Bryant, Joy D Humphreys, Glenn E Smith, Robert J Ivnik, Neill R Graff-Radford, Ronald C Petersen, and John A Lucas. Detecting dementia with the minimal state examination in highly educated individuals. *Archives of neurology*, 65(7):963–967, 2008.
- [43] Sebastian Palmqvist, Pontus Tideman, Nicholas Cullen, Henrik Zetterberg, Kaj Blennow, Alzheimer's Disease Neuroimaging Initiative, Jeffery L Dage, Erik Stomrud, Shorena Janelidze, Niklas Mattsson-Carlgen, et al. Prediction of future alzheimer's disease dementia using plasma phospho-tau combined with other accessible measures. *Nature medicine*, 27(6):1034–1042, 2021.
- [44] Lucilla Parnetti, Elena Chipi, Nicola Salvadori, Katia D'Andrea, and Paolo Eusebi. Prevalence and risk of progression of preclinical alzheimer's disease stages: a systematic review and meta-analysis. *Alzheimer's research & therapy*, 11:1–13, 2019.
- [45] Ronald Carl Petersen, Paul S Aisen, Laurel A Beckett, Michael C Donohue, Anthony Collins Gamst, Danielle J Harvey, CR Jack Jr, William J Jagust, Leslie M Shaw, Arthur W Toga, et al. Alzheimer's disease neuroimaging initiative (adni) clinical characterization. *Neurology*, 74(3):201–209, 2010.
- [46] Carlos Platero. Categorical predictive and disease progression modeling in the early stage of alzheimer's disease. *Journal of Neuroscience Methods*, 374:109581, 2022.
- [47] Carlos Platero. Temporal modeling and at profiles in the early phase of alzheimer's disease. *Journal of Alzheimer's Disease Reports*, 9:25424823241306097, 2025.
- [48] Sociedad Española de Neurología. La sen advierte que entre un 30 y un 40 % de los casos de demencia se podrían prevenir, 2017.
- [49] Janina Ulbl and Martin Rakusa. The alzheimer's disease assessment scale–cognitive subscale (adas-cog): Validation of slovenian version for detecting mild cognitive impairment or alzheimer's dementia. *Alzheimer's & Dementia*, 17:e056177, 2021.
- [50] Lei Xu, Hui Wu, Chunming He, Jun Wang, Changqing Zhang, Feiping Nie, and Lei Chen. Multi-modal sequence learning for alzheimer's disease progression prediction with incomplete variable-length longitudinal data. *Medical Image Analysis*, 82:102643, 2022.

- [51] Hao Yu. Rmpi: Interface (Wrapper) to MPI (Message-Passing Interface). CRAN, 2023. R package version 0.7-2.