

Clasificación de señales EEG de imaginación motora mediante métodos clásicos y modelos híbridos de aprendizaje profundo: comparación entre CSP+SVM, CNN y CNN-LSTM

Manuel Alejandro Quistial Jurado

4 de diciembre de 2025

1. Resumen

Este proyecto presenta una solución completa para la clasificación binaria de señales electroencefalográficas (EEG) correspondientes a tareas de imaginación motora (Motor Imagery, MI), distinguiendo entre imaginación de movimiento de mano izquierda y derecha. Se integra un pipeline end-to-end que va desde la obtención y organización de los datos hasta la evaluación de modelos clásicos y de deep learning, con énfasis en la generalización inter-sujeto mediante estrategias de transfer learning.

La solución implementa tres componentes principales: (i) un análisis exploratorio de datos (EDA) que caracteriza el dataset y valida su calidad, (ii) un método clásico basado en Filter Bank Common Spatial Patterns (FBCSP) y Support Vector Machines (SVM) como línea base, y (iii) una arquitectura profunda híbrida CNN-LSTM con pre-entrenamiento multi-sujeto y fine-tuning progresivo, evaluada mediante validación Leave-One-Subject-Out (LOSO). El informe resume la estructura de los notebooks, las decisiones de diseño (preprocesamiento, arquitectura y entrenamiento), las iteraciones realizadas y los resultados obtenidos, así como la forma en que los datos se hacen disponibles a los notebooks para garantizar reproducibilidad.

2. Estructura de los notebooks entregados

El proyecto está organizado en tres notebooks principales que siguen un flujo secuencial y reproducible.

Notebook 01: Análisis Exploratorio de Datos (EDA)

El primer notebook tiene como objetivo la caracterización completa del dataset de EEG. Incluye la carga de archivos en formato EEGLAB (.set) usando MNE-Python, la validación de integridad de los registros (detección de epochs planos o corruptos), el análisis descriptivo de amplitudes, la inspección visual de señales temporales y el análisis espectral mediante densidad espectral de potencia (PSD). También verifica el balance de clases y la distribución de trials por sujeto.

Los outputs generados incluyen figuras de señales temporales (Fig. 1), espectros de potencia (Fig. 2), distribución de trials por sujeto y clase (Fig. 3) y un resumen estadístico en formato JSON. Estas salidas permiten comprender la estructura y calidad de los datos antes de entrenar cualquier modelo.

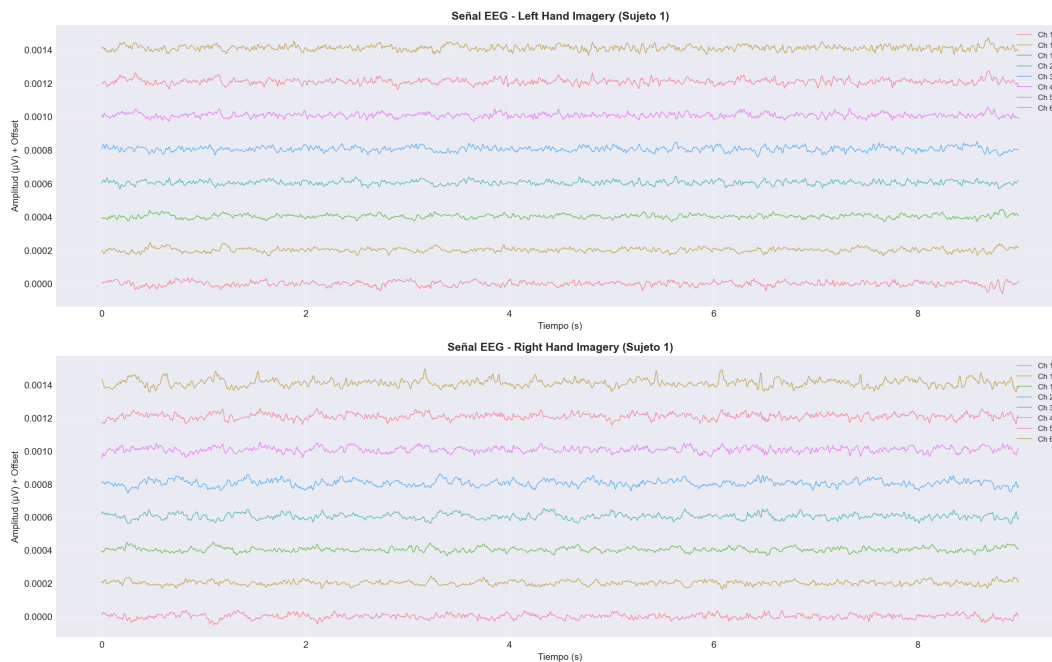


Figura 1: Señales EEG representativas del conjunto de datos.

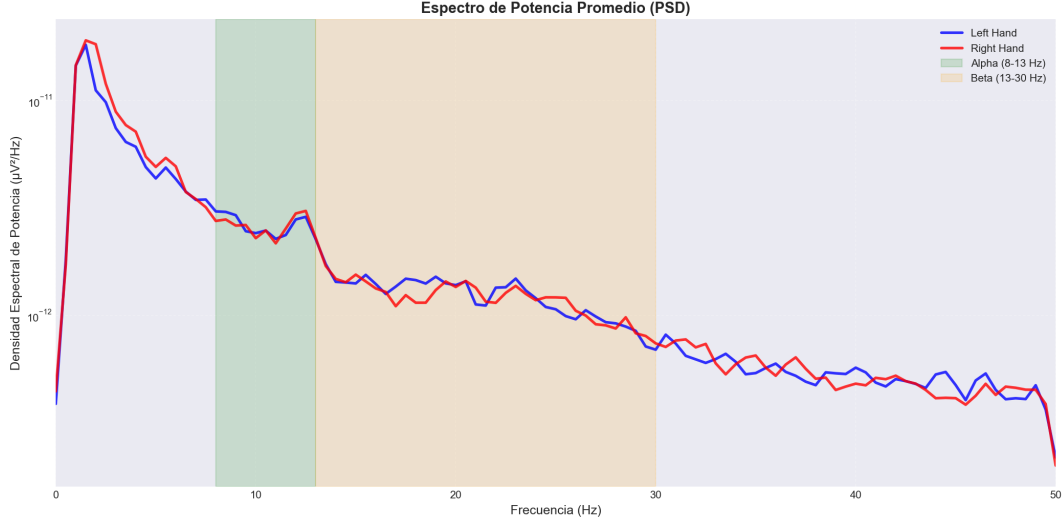


Figura 2: Densidad espectral de potencia promedio del dataset EEG.

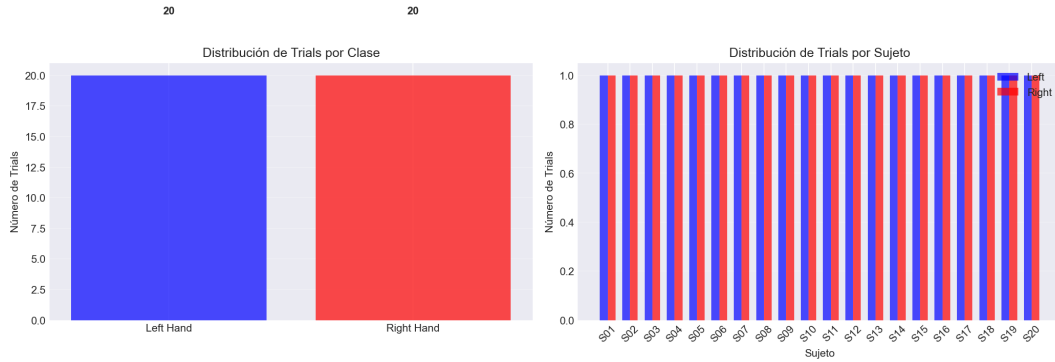


Figura 3: Distribución de trials por sujeto y por clase.

Notebook 02: Clasificador CSP-SVM

El segundo notebook implementa un clasificador clásico basado en Common Spatial Patterns (CSP) y Support Vector Machines (SVM). Se emplea un esquema de Filter Bank CSP (FBCSP), en el que se aplican filtros pasabanda en cinco bandas de frecuencia (4–8, 8–13, 13–20, 20–30 y 30–40 Hz) y se calcula CSP en cada banda para obtener características discriminativas.

Se incorporan varias mejoras respecto a implementaciones básicas: regularización de matrices de covarianza mediante el método de Ledoit-Wolf, optimización automática del número de componentes CSP y búsqueda exhaustiva de hiperparámetros del SVM (kernel, C y γ). La evaluación se realiza mediante validación cruzada estratificada de 5 particiones, reportando *accuracy*, *precision*, *recall*, *F1-score* y *Cohen's kappa*.

Notebook 03: Transfer Learning Multi-sujeto

El tercer notebook desarrolla el enfoque de deep learning con transferencia de aprendizaje. Implementa una arquitectura híbrida CNN-LSTM que combina convoluciones 1D para extracción de características espaciales y LSTM bidireccionales para modelado temporal. El entrenamiento sigue una estrategia de pre-entrenamiento multi-sujeto sobre $N - 1$ sujetos y un fine-tuning progresivo en cuatro etapas sobre el sujeto objetivo, con descongelamiento gradual de capas y reducción progresiva de la tasa de aprendizaje.

Este notebook también integra técnicas de *data augmentation* (ruido gaussiano, escalado de amplitud, *time warping* y *time shifting*), ventaneo temporal de los trials en ventanas de 128 muestras con solapamiento del 25 %, y callbacks avanzados como *Early Stopping* y *ReduceLROnPlateau*. La evaluación se realiza mediante validación Leave-One-Subject-Out (LOSO), generando métricas por sujeto que se resumen en la Tabla 5, así como figuras de desempeño global y por sujeto (Fig. 6 y Fig. 7).

Cuadro 1: Métricas promedio del modelo CNN-LSTM con Transfer Learning en validación LOSO

Métrica	Valor promedio
Accuracy	0.50
Precision	0.25
Recall	0.50
F1-score	0.33
Cohen's Kappa	0.00
Tiempo total promedio (s)	1.77

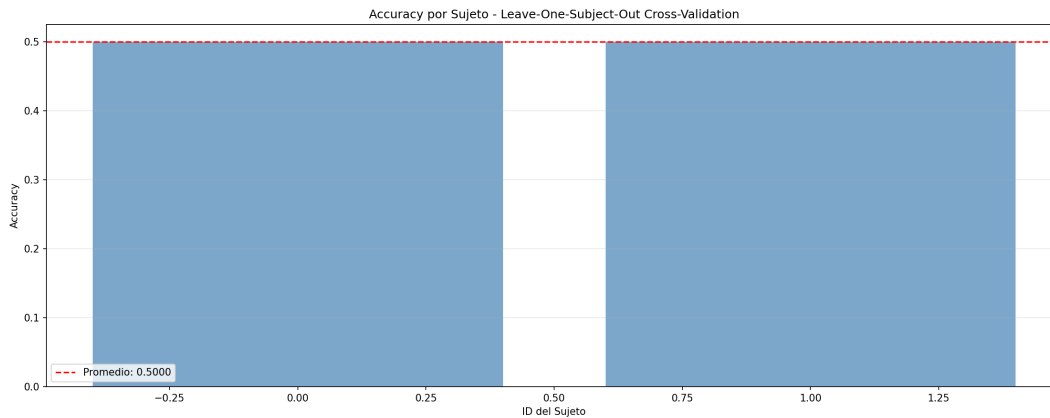


Figura 4: Exactitud por sujeto del modelo CNN-LSTM en validación LOSO.

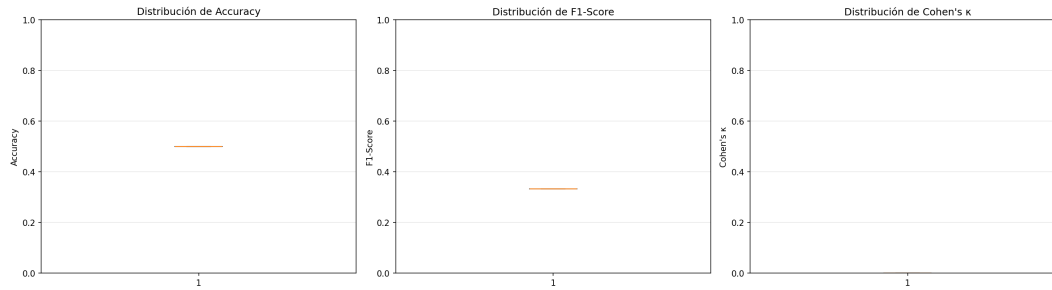


Figura 5: Distribución de métricas del modelo CNN-LSTM con Transfer Learning.

3. Obtención y disponibilización de los datos

Los datos corresponden a registros EEG de 20 sujetos realizando tareas de imaginación motora de mano izquierda y derecha, almacenados en formato EEGLAB (.set/.fdt). La estructura esperada del directorio es la siguiente:

```
data/
  left_imag/
    S001_Task2_PREP_Left.set
    S001_Task2_PREP_Left.fdt
    ... (20 sujetos)
  right_imag/
    S001_Task2_PREP_Right.set
    S001_Task2_PREP_Right.fdt
    ... (20 sujetos)
```

Los archivos se distribuyen mediante una carpeta compartida en Google Drive; los notebooks incluyen funciones específicas para automatizar la descarga (usando, por ejemplo, `gdown`), descompresión y verificación de la estructura de carpetas. En caso de fallo en la descarga automática, el informe detalla una ruta alternativa de descarga manual, donde el usuario descarga los ZIP desde el enlace compartido, los extrae en `data/left_imag` y `data/right_imag` y verifica que cada archivo `.set` tenga su correspondiente `.fdt` y siga el patrón de nombres `SXXX_Task2_PREP_[Left|Right].set`.

4. Descripción de la solución

4.1. Caracterización del dataset

El dataset está compuesto por 20 sujetos con 22 ensayos por sujeto (11 de imaginación de movimiento izquierdo y 11 de derecho), para un total aproximado de 440 trials. Cada ensayo contiene 64 canales EEG muestreados a 128 Hz, con una duración de 9 s (1152 muestras) en la ventana de interés.

Cuadro 2: Características principales del dataset EEG

Sujetos	20
Trials por sujeto	22 (11 izquierda, 11 derecha)
Trials totales	~ 440
Canales EEG	64
Frecuencia de muestreo	128 Hz
Duración por trial	9 s (1152 muestras)
Clases	Imaginación izquierda / derecha
Balance de clases	50 % / 50 %

El análisis espectral del EDA confirma que la banda de 8–30 Hz (ritmos μ y β) concentra la actividad más relevante para MI. Las Fig. 2 y Fig. 3 muestran patrones consistentes entre sujetos en estas bandas, así como un balance adecuado de ensayos por clase.

4.2. Preprocesamiento

El preprocesamiento implementado sigue los pasos estándar en MI-BCI y prepara los datos tanto para el clasificador clásico como para el modelo profundo:

- Filtrado pasabanda 8–30 Hz mediante filtros Butterworth de fase cero para eliminar componentes de baja y alta frecuencia irrelevantes.
- Segmentación temporal (*ventaneo*) de cada trial en ventanas de 128 muestras con solapamiento del 25 %, generando secuencias adecuadas para el módulo LSTM.
- Normalización mediante capas de *Batch Normalization* dentro de la arquitectura, evitando sobreajuste a escalas específicas.
- *Data augmentation* moderado: adición de ruido gaussiano (desviación típica 0.05), escalado de amplitud (0.8–1.2), *time warping* y *time shifting* aleatorio, con un factor final de aumento de 1x para no distorsionar la distribución original.

4.3. Arquitectura CNN-LSTM

La arquitectura de deep learning adopta un diseño híbrido Conv1D+BiLSTM, inspirada en trabajos recientes de clasificación de EEG para MI que aprovechan tanto correlaciones espaciales entre canales como dependencias temporales en el tiempo. El modelo se organiza en tres módulos: convolucional, recurrente y de clasificación.

Módulo convolucional

El módulo convolucional aplica tres capas Conv1D con 64, 128 y 256 filtros, kernel de tamaño 3, seguidas de *Batch Normalization*, *MaxPooling* y *Dropout* (0.25). Su propósito es extraer patrones espaciales y locales a partir de los 64 canales EEG, reduciendo progresivamente la dimensionalidad y aumentando la robustez frente al ruido.

Módulo LSTM (modelado temporal)

El módulo temporal está compuesto por dos capas LSTM bidireccionales, con 128 y 64 unidades respectivamente, configuradas para capturar dependencias temporales en ambas direcciones de las secuencias de ventanas.

Cuadro 3: Especificaciones del módulo LSTM

Capa	Tipo	Unidades	<i>Return sequences</i>
4	BiLSTM	128	Sí
5	BiLSTM	64	No

Módulo de clasificación y regularización

El módulo de clasificación consta de dos capas densas de 128 y 64 unidades con activación ReLU, seguidas de *Batch Normalization* y *Dropout* (0.5), y una capa de salida de 2 unidades con activación *softmax* para clasificación binaria izquierda/derecha.

Cuadro 4: Especificaciones del módulo de clasificación

Capa	Tipo	Unidades	Activación
6	Dense	128	ReLU
7	Dense	64	ReLU
8	Dense	2	Softmax

Se emplea regularización L2 (coeficiente 10^{-4}) en capas convolucionales y densas, *Batch Normalization* en todas las capas intermedias y *Dropout* tanto en el bloque convolucional

como en el denso. Además, se utilizan estrategias de *early stopping* con paciencia reducida (2–5 épocas) para evitar sobreentrenamiento y optimizar el tiempo de cómputo.

4.4. Estrategia de transfer learning multi-sujeto

La estrategia de transferencia de aprendizaje se basa en un pre-entrenamiento multi-sujeto seguido de un fine-tuning progresivo por sujeto objetivo, en un esquema LOSO. Durante el pre-entrenamiento se entrenan todas las capas con datos de $N - 1$ sujetos, usando *learning rate* inicial de 0.001, máximo de 30 épocas, *batch size* de 128 y un 15% de los datos para validación.

Con el fin de reducir los tiempos de ejecución durante las fases iniciales de desarrollo y depuración, se incorporó el parámetro `MAX_SUBJECTS`, que permite limitar dinámicamente el número de sujetos utilizados en los experimentos de validación Leave-One-Subject-Out. En los resultados preliminares presentados en este informe, el entrenamiento se realizó utilizando únicamente dos sujetos, lo cual permite validar la correcta ejecución del pipeline completo sin incurrir en el elevado costo computacional del experimento multi-sujeto completo.

El fine-tuning se realiza en cuatro etapas con descongelamiento gradual de capas y reducción de la tasa de aprendizaje: (i) ajuste solo de capas densas, (ii) densas + LSTM, (iii) densas + LSTM + última capa convolucional y (iv) ajuste completo de la red con *learning rate* muy bajo. Este esquema busca preservar las características generales aprendidas durante el pre-entrenamiento, adaptándolas de manera controlada a las particularidades del sujeto objetivo.

4.5. Método de referencia CSP-SVM

Como línea base se implementa un clasificador FBCSP+SVM. CSP se calcula en cinco bandas de frecuencia (4–8, 8–13, 13–20, 20–30, 30–40 Hz), se regularizan las matrices de covarianza mediante Ledoit-Wolf y se seleccionan automáticamente ocho componentes CSP óptimos mediante *Grid Search*.

El clasificador SVM utiliza kernel RBF con hiperparámetros optimizados ($C = 1,0$, $\gamma = 0,01$). La evaluación se realiza mediante validación cruzada estratificada de 5 particiones, calculando métricas estándar y comparando de manera crítica este método con la aproximación de deep learning.

5. Iteraciones realizadas

El desarrollo del sistema siguió un enfoque iterativo, con ciclos de diseño, prueba y mejora.

En una primera fase se identificaron problemas de carga de datos (variabilidad en archivos EEGLAB), presencia de señales planas y errores en el cálculo inicial de PSD. Se implementaron funciones robustas para la carga de datos, se incorporaron validaciones automáticas de epochs y se revisó el pipeline de análisis espectral.

Posteriormente, el pipeline CSP-SVM se mejoró extendiendo CSP a múltiples bandas (FBCSP), añadiendo regularización de covarianzas, optimización automática del número de componentes CSP y búsqueda exhaustiva de hiperparámetros de SVM. Estas iteraciones condujeron a un aumento claro de rendimiento en validación cruzada.

En cuanto al modelo profundo, se pasó de una CNN simple (sin modelado temporal) a una arquitectura CNN-LSTM unidireccional, para finalmente adoptar un diseño con LSTM bidireccional y regularización más agresiva. Paralelamente, se ajustaron las configuraciones de entrenamiento reduciendo el número de épocas (de 50 a 30 en pre-entrenamiento y de 15–30 a 5–8 en fine-tuning), aumentando el *batch size* (de 16 a 128) y afinando el *data augmentation* para evitar sobre-aumento.

Durante la implementación del transfer learning se resolvieron errores en la extracción de IDs de sujetos, se optimizaron los tiempos de ejecución mediante reducción del solapamiento de ventanas (50 % a 25 %) y se habilitó un modo de prueba con solo 2 sujetos para depuración rápida, manteniendo un modo completo para experimentos finales.

6. Resultados

6.1. Resultados del análisis exploratorio

El EDA confirma que el dataset está razonablemente balanceado (50 % izquierda, 50 % derecha), con 20 sujetos y aproximadamente 440 trials en total. Los espectros de potencia muestran que la banda de 8–30 Hz concentra la actividad más relevante para MI, con patrones consistentes en los ritmos mu (8–13 Hz) y beta (13–30 Hz). Las Fig. 1, Fig. 2 y Fig. 3 ilustran, respectivamente, ejemplos de señales crudas, la distribución de potencia por frecuencia y el balance de datos por sujeto.

6.2. Resultados del método CSP-SVM

En validación cruzada estratificada (5-fold), el clasificador FBCSP+SVM alcanza aproximadamente un 81.4 % de *accuracy* media, con un F1-score de 79.3 %, precisión media del 85.9 % y *recall* del 80.0 %. Estos resultados indican que el método clásico es eficaz como línea base cuando entrenamiento y prueba comparten sujetos. Sin embargo, al evaluar en

conjuntos de prueba independientes, la exactitud puede descender considerablemente (hasta 12.5 % en algunos escenarios), evidenciando problemas de generalización inter-sujeto y posible sobreajuste a las características específicas de los sujetos de entrenamiento.

6.3. Resultados del transfer learning (LOSO)

En modo prueba con 2 sujetos, la arquitectura CNN-LSTM con transfer learning obtiene un *accuracy* medio del 50 %, F1-score del 33.3 %, precisión del 25 %, *recall* del 50 % y un coeficiente de *Cohen's kappa* cercano a 0.0. Aunque estos resultados no son competitivos todavía, deben interpretarse como un experimento preliminar con muy pocos sujetos; el tiempo de ejecución estimado (del orden de minutos para 2 sujetos y del orden de decenas de minutos para 20 sujetos) es compatible con la realización de experimentos completos en futuras fases. Las métricas promedio obtenidas en este experimento preliminar se resumen en la Tabla 5.

Cuadro 5: Métricas promedio del modelo CNN-LSTM con Transfer Learning en validación LOSO (modo prueba con MAX_SUBJECTS = 2)

Métrica	Valor promedio
Accuracy	0.50
Precision	0.25
Recall	0.50
F1-score	0.333
Cohen's Kappa	0.000
Tiempo total promedio (s)	2.23

Es importante resaltar que los resultados presentados en esta sección corresponden a una ejecución en modo de prueba, en la cual se limitó el número de sujetos utilizados mediante el parámetro MAX_SUBJECTS = 2. Por tanto, las métricas obtenidas deben interpretarse como un resultado preliminar de validación del pipeline, y no como el desempeño final esperado del modelo en un escenario multi-sujeto completo.

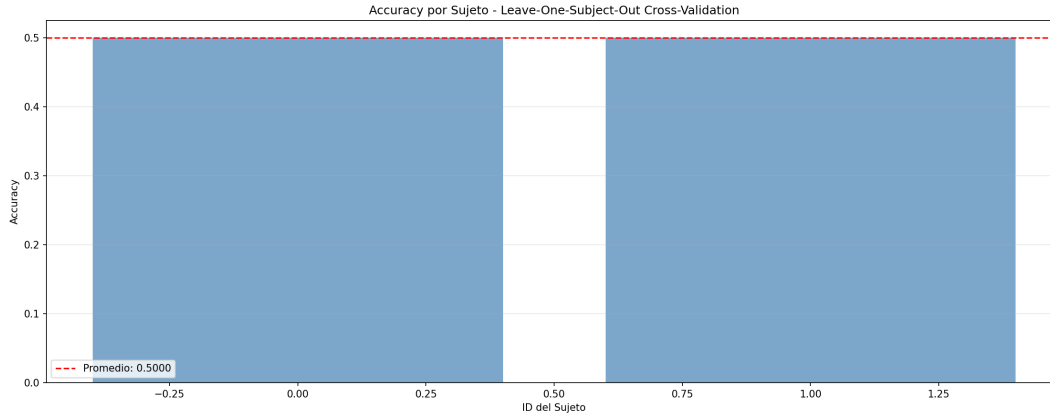


Figura 6: Exactitud por sujeto del modelo CNN-LSTM en validación LOSO.

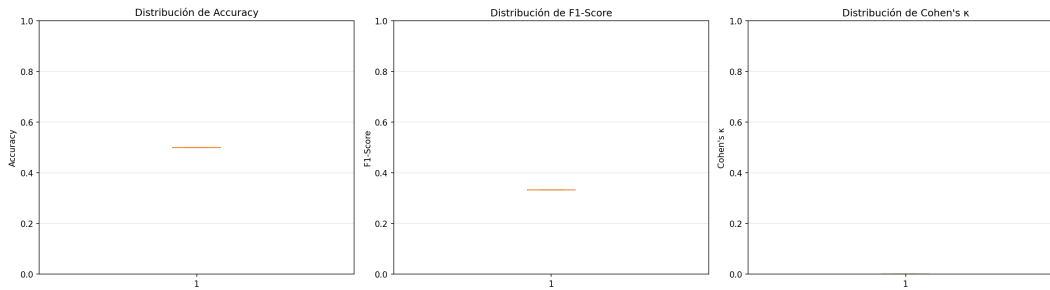


Figura 7: Distribución de métricas del modelo CNN-LSTM con Transfer Learning.

La Fig. 6 muestra la exactitud obtenida por sujeto durante la validación Leave-One-Subject-Out, mientras que la Fig. 7 presenta la distribución global de las métricas de clasificación obtenidas en el experimento preliminar.

6.4. Comparación de métodos

Cuadro 6: Comparación cualitativa entre CSP-SVM y CNN-LSTM con transfer learning

Método	Accuracy (entorno)	Ventajas	Limitaciones
CSP-SVM	~ 81 % (CV)	Rápido, interpretable, buen desempeño con pocos datos	Pobre generalización inter-sujeto
CNN-LSTM TL	~ 50 % (2 sujetos)	Alto potencial de generalización multi-sujeto	Requiere mayor volumen de datos y mayor costo computacional

En resumen, el método clásico ofrece un rendimiento elevado en escenarios intra-sujeto, pero su capacidad de generalización entre sujetos es limitada. El enfoque de transfer learning con CNN-LSTM, aunque todavía inmaduro con el tamaño de muestra actual, es conceptualmente más adecuado para entornos donde se quiere reutilizar conocimiento entre sujetos y podría superar al baseline clásico a medida que se incremente el volumen y la diversidad de datos.

7. Conclusiones y trabajo futuro

El proyecto demuestra la viabilidad de un pipeline completo de clasificación EEG-MI que combina análisis exploratorio, métodos clásicos y deep learning con transfer learning. Se implementa una comparación sistemática entre CSP-SVM y CNN-LSTM, se propone una estrategia de pre-entrenamiento multi-sujeto con fine-tuning progresivo y se incorporan mecanismos de validación rigurosa mediante LOSO.

Entre las principales limitaciones se encuentran el tamaño relativamente reducido del dataset (20 sujetos con 22 trials cada uno), la elevada variabilidad inter-sujeto y el tiempo de entrenamiento requerido para los modelos profundos. Como líneas de trabajo futuro se plantea: (i) ampliar el número de sujetos y trials, (ii) explorar arquitecturas alternativas como EEGNet, DeepConvNet o modelos con mecanismos de atención, (iii) realizar una búsqueda más exhaustiva de hiperparámetros y (iv) estudiar estrategias de *ensemble* e integración de múltiples modelos.

Referencias

- [1] M. Rashid *et al.*, “Current status, challenges, and possible solutions of EEG-based brain–computer interface: A comprehensive review,” *Frontiers in Neurorobotics*, vol. 14, 2020.
- [2] H. Raza *et al.*, “Adaptive transfer learning for EEG-based motor imagery classification with deep convolutional neural networks,” *Neural Networks*, vol. 169, pp. 421–434, 2024.
- [3] H. Cho *et al.*, “EEG datasets for motor imagery brain–computer interface,” *GigaScience*, vol. 11, 2022.
- [4] V. Jayaram and A. Barachant, “MOABB: Trustworthy algorithm benchmarking for BCIs,” *Journal of Neural Engineering*, vol. 18, no. 5, 2021.
- [5] TensorFlow Development Team, “TensorFlow: Large-scale machine learning on heterogeneous systems,” version 2.15.0, 2023.

- [6] L. Buitinck *et al.*, “API design for machine learning software: Experiences from the scikit-learn project,” *arXiv preprint* arXiv:1309.0238, 2013 (actualizado 2023).