



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**ANÁLISIS LONGITUDINAL DE ELECTROENCEFALOGRAMAS DE
PACIENTES ESQUIZOFRÉNICOS MEDIANTE APRENDIZAJE DE
MÁQUINAS**

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA,
MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL ELÉCTRICA

FRANCISCA SALOMÉ CONA FLORES

PROFESOR GUÍA:
PABLO ESTÉVEZ VALENCIA

MIEMBROS DE LA COMISIÓN:
JOSÉ CORTÉS BRIONES
CESAR AZURDIA MEZA

Este trabajo ha sido parcialmente financiado por:
ANID – Programa Iniciativa Científica Milenio – ICN2021_004 y Fondecyt 1220829

SANTIAGO DE CHILE
2023

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL ELÉCTRICA
POR: FRANCISCA SALOMÉ CONA FLORES
FECHA: 2023
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

ANÁLISIS LONGITUDINAL DE ELECTROENCEFALOGRAMAS DE PACIENTES ESQUIZOFRÉNICOS MEDIANTE APRENDIZAJE DE MÁQUINAS

La esquizofrenia corresponde a una enfermedad mental crónica que posee un fuerte efecto disruptivo en el círculo y calidad de vida de los pacientes. Son diversos los estudios que abordan la búsqueda de biomarcadores en registros encefalográficos (EEG), con los cuales se ha logrado discriminar entre pacientes esquizofrénicos y sujetos sanos.

En esta tesis se propone discriminar entre registros EEG, de un mismo paciente, con respecto a la intensificación de su sintomatología. Se dispone de una base de datos pionera, conformada por registros EEG de 21 pacientes monitoreados regularmente; cada registro es acompañado de evaluaciones sintomatológicas. Se desarrolla una *pipeline* automatizada de preprocesamiento para limpiar los registros. Posteriormente, se analizan características extraídas manualmente, para identificar patrones en gráficas de dispersión y modelos preliminares. Se implementa un modelo de aprendizaje profundo que utiliza transferencia de aprendizaje, a partir de otra tarea relacionada a esquizofrenia, y logra obtener resultados relativamente efectivos. Se evalúa el modelo parcialmente, ya que la tarea debe ajustarse para cada paciente, y la cantidad de registros encefalográficos es reducida. Esta tesis entrega un primer acercamiento al monitoreo de la intensificación de los síntomas de pacientes con esquizofrenia analizando EEGs; esto se considera un buen *baseline* para futuros estudios.

*A mi mamá, papá, hermanos, a mi pareja, y a mis amados perros y gata,
gracias por el apoyo permanente e incondicional.*

“It always seems impossible, until it’s done” - Nelson Mandela

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Hipótesis	2
1.3. Objetivo general	3
1.4. Objetivos específicos	3
1.5. Organización del documento	3
2. Marco teórico	5
2.1. Esquizofrenia	5
2.2. Electroencefalograma (EEG) y su aplicación en EQZ	5
2.2.1. Artefactos y ruido en el EEG	8
2.3. Complejidad de Lempel-Ziv	9
2.4. Aprendizaje de máquinas	11
2.4.1. Redes neuronales artificiales	12
2.4.2. Aprendizaje supervisado	13
2.4.3. Entrenamiento por retropropagación del error	14
2.4.4. Aprendizaje profundo	15
2.4.5. Tipos de capas neuronales	16
2.4.6. Tipos de funciones de activación	17
2.4.7. Transferencia de aprendizaje	18
2.4.8. Cuidados con el gradiente en el entrenamiento	19
2.4.9. Cuidados con la generalización del modelo	20
2.4.10. <i>Random Forest</i>	20
3. Metodología	22
3.1. Descripción general del problema	22
3.1.1. Datos	22
3.1.2. Tarea a resolver	22
3.1.3. Solución propuesta	22
3.2. Base de datos	23
3.2.1. Limpieza de las señales	25
3.2.2. Etiquetado de las señales	29
3.2.3. Base de datos auxiliar	32
3.3. Enfoque con extracción manual de características	33
3.3.1. Características a extraer	33
3.3.2. Extracción y análisis de características	37
3.4. Enfoque con aprendizaje profundo	38

3.4.1.	Arquitectura	38
3.4.1.1.	Preprocesamiento de la señal	41
3.4.2.	Pesos de preentrenamiento	43
3.4.3.	Aumento de datos (<i>Data augment</i>)	43
3.4.4.	Evaluación del modelo	44
3.4.5.	Hiperparámetros del modelo	46
3.5.	Softwares utilizados	47
4.	Resultados	48
4.1.	Enfoque con extracción manual de características	48
4.2.	Enfoque con aprendizaje profundo	53
4.2.1.	Primera aproximación con datos auxiliares	53
4.2.2.	Resultados con base de datos <i>DecompStudy</i>	55
5.	Discusión	67
5.1.	Enfoque con extracción manual de características	67
5.2.	Enfoque con aprendizaje profundo	68
5.2.1.	Primera aproximación con datos auxiliares	68
5.2.2.	Resultados con base de datos <i>DecompStudy</i>	69
6.	Conclusión	72
6.1.	Trabajo a futuro	73
	Bibliografía	75
	Anexos	80
A.	Ilustración criterio de etiquetado por sujeto	81
B.	<i>Sanity checks</i> LZC	81

Índice de Tablas

3.1.	Información general acerca de los sujetos participantes de <i>DecompStudy</i>	25
3.2.	Hiperparámetros para el conjunto de datos <i>EQZvsHC</i>	46
4.1.	Resultados de <i>random forest</i> utilizando el conjunto de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] como entrada.	52
4.2.	Resultados de <i>random forest</i> utilizando el conjunto de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] como entrada.	52
5.1.	Detalles de los registros de validación mal clasificados según las figuras 4.20-4.26.	70

Índice de Ilustraciones

2.1.	Muestra de EEG normal de 20 canales acompañado de un canal de electrocardiograma; registrado durante <i>wakefulness</i> en una mujer de 24 años. Obtenido de [17].	6
2.2.	Ejemplo de ubicación de electrodos usando ya sea un sistema 10-20 (círculos negros), o un sistema combinacional modificado (círculos blancos + círculos negros)[16].	7
2.3.	Ejemplo de una señal EEG observada que incluye artefactos biológicos. Figura obtenida de [23].	9
2.4.	Ejemplo gráfico de cálculo de LZC, para segmento de una señal.	10
2.5.	Funciones de activación más utilizadas en las arquitecturas de redes neuronales.	18
2.6.	Ilustración de árbol de decisión simple para decidir si salir a jugar tennis o no. Las hojas son las 4 opciones presentes al inferior de la imagen, y corresponden a los valores que puede tomar la variable <i>target</i> . Las ramas están representadas por los recuadros.	21
3.1.	5 segundos de una señal EEG luego de aplicar la limpieza de datos ASR (línea roja) superpuesta en la señal EEG original (línea azul).	27
3.2.	5 segundos de una señal EEG justo antes de ingresar a la red neuronal del paso 4 (línea azul), y justo después de salir del re-referenciado del paso 7 (línea roja).	29
3.3.	Puntajes $SAPS_{global}$ y $PANSS_{positive}$ reportados por los metadatos gráficos en puntos. Mientras que se sobrepone las rectas generadas por las ecuaciones $\widehat{SAPS}_{global} = -3.222 + (0.567 * PANSS_{positive})$ (recta anaranjada) y $PANSS_{positive} = 9.3264 + (1.1072 * SAPS_{global})$ (recta azul), introducidas por análisis regresivo en [51].	31
3.4.	Ilustración del exceso de <i>kurtosis</i> en una distribución.	34
3.5.	Ilustración del <i>skewness</i> de una distribución.	34
3.6.	Ilustración de umbral usado en cada método de binarización previo al cómputo de la LZC. La línea punteada roja indica el umbral considerado para que los valores en azul sean codificados como ‘1’s y los valores en verde sean codificados como ‘0’s. Notar que en cada gráfico se plotea la misma señal pero a diferente escala, con el fin de observar mejor los criterios utilizados para establecer el umbral.	36
3.7.	Arquitectura del modelo propuesto para aprendizaje profundo. De izquierda a derecha con líneas punteadas se indican los bloques: Codificador, <i>Encoding augment</i> y <i>Extended classifier</i>	41
3.8.	Un ejemplo de cómo se ve un segmento justo antes de ingresar a la red. Es decir, un segmento previamente limpiado y preprocesado.	42

3.9.	Método utilizado en cada <i>batch</i> , durante el entrenamiento, para regularizar el entrenamiento.	44
3.10.	Ilustración del criterio utilizado para computar los valores de las curvas de validación. Se ilustra con un canal, pero la misma idea se extiende a los 20 canales que se usan.	46
4.1.	Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.	48
4.2.	Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.	49
4.3.	Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.	49
4.4.	Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.	49
4.5.	Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.	50
4.6.	Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.	50
4.7.	Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.	50
4.8.	Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.	51
4.9.	Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.	51
4.10.	Muestra de uno de los árboles de decisión resultante para el sujeto SA007. . .	53
4.11.	Proyección 2D de la representación de segmentos de 40 s en el espacio latente obtenida por el codificador convolucional, para la base de datos <i>EQZvsHC</i> . . .	54
4.12.	Comparación de las curvas de entrenamiento/validación al modificar la inicialización de los pesos, entrenando con los datos <i>EQZvsHC</i> . Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Curvas rojo-naranja corresponden a una inicialización por defecto, mientras que las curvas rosa-morado corresponde a la utilización de transferencia de aprendizaje.	54

4.13.	Comparación de las curvas de entrenamiento/validación sin congelar parámetros (curvas rojo-naranja) y con congelamiento del inicio de la red (rojas rosa-morado), entrenando con los datos <i>EQZvsHC</i> . Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b).	55
4.14.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA007 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	56
4.15.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA010 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	56
4.16.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA014 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	57
4.17.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA025 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	57
4.18.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA026 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	58
4.19.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA039 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	58
4.20.	Comparación de curvas de entrenamiento y validación al entrenar con el sujeto SA047 con diferentes inicializaciones de parámetros. Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.	59

4.21.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA007 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	60
4.22.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA010 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	61
4.23.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA014 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	62
4.24.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA025 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	63
4.25.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA026 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	63
4.26.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA039 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	64
4.27.	Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de <i>EQZvsHC</i> , para registros EEG del sujeto SA047 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación. .	64
4.28.	Curva de entrenamiento y validación al entrenar con el sujeto SA011 . Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b).	65
4.29.	Predicciones del modelo entrenado con pesos iniciales de <i>EQZvsHC</i> , para registros EEG del sujeto SA011 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.	65
4.30.	Curva de entrenamiento y validación al entrenar con el sujeto SA012 . Curvas de <i>accuracy</i> en (a) y curvas de <i>loss</i> en (b).	66
4.31.	Predicciones del modelo entrenado con pesos iniciales de <i>EQZvsHC</i> , para registros EEG del sujeto SA012 . Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.	66

A.1.	Ilustración del proceso utilizado para etiquetar las señales EEG para cada sujeto; en cada recuadro está representada la recta de la ec. (3.2) por una línea punteada gris. Los puntos azules corresponden a los valores explícitos $SAPS_{global}$ y $PANSS_{positive}$, según los metadatos registrados. Las estrellas anaranjadas representan el valor del \widehat{SAPS}_{global} estimado dado el valor $PANSS_{positive}$ registrado en los metadatos. Luego, las cruces rojas representan el valor $SAPS_{def}$, es decir, el valor promedio entre $SAPS_{global}$ y \widehat{SAPS}_{global} . Finalmente, la línea punteada roja representa el umbral binarizador, dado por la mediana entre los valores \widehat{SAPS}_{global} de cada sujeto.	81
B.1.	Señales generadas y su respectivo valor calculado de LZC -median.	82
B.2.	Señales generadas a las cuales se les computa la LZC por ventana deslizante de 10 s.	84
B.3.	Evolución del valor de LZC por ventana deslizante (10 s con traslape del 90% entre sí) a lo largo de las señales descritas en B.2.	84

1. Introducción

1.1. Motivación

La esquizofrenia (EQZ) es una enfermedad mental crónica que afecta aproximadamente al 0.45 % de la población mundial adulta [1]. La EQZ es un síndrome complejo y heterogéneo asociado a alteraciones graves en la percepción (e.g., alucinaciones auditivas verbales), cognición (e.g., memoria de trabajo reducida) y emoción (e.g., aplanamiento afectivo). Si bien sus causas no están del todo claras, se cree que la EQZ es un trastorno del desarrollo que involucra tanto factores genéticos como ambientales.

Más específicamente, dentro de los síntomas de la EQZ se encuentran: alucinaciones verbales, delirios, desórdenes formales del pensamiento, aplanamiento afectivo y abulia, entre otros. Es por lo anterior que la EQZ tiene un efecto disruptivo profundo tanto en los pacientes, como en sus cuidadores, familiares y la sociedad que los rodea. Más del 50 % de los individuos diagnosticados presentan episodios de descompensación clínica de manera intermitente, y alrededor del 20 % presentan síntomas crónicos y discapacidad [2].

El tratamiento consiste, principalmente, en medicación antipsicótica combinada con terapia psicológica, apoyo social y rehabilitación. La medicación es efectiva reduciendo o eliminando alucinaciones, delirios y desórdenes de pensamiento en un porcentaje importante de los pacientes. En pacientes clínicamente compensados, la sintomatología puede llegar a reducirse hasta niveles que permiten a muchos pacientes recuperar parte de su independencia y autonomía. Sin embargo, es frecuente que los estados de compensación clínica sean interrumpidos por episodios de descompensación, i.e., empeoramiento significativo de la sintomatología, que, usualmente, requiere de la hospitalización de los pacientes. Estos episodios no solo interrumpen la vida diaria de los pacientes y sus cercanos, sino que, además, se asocian a peores pronósticos, menor calidad de vida, desempleo, indigencia y problemas generales de salud. Prevenir o interrumpir de manera temprana el curso de los episodios descompensación por medio de intervenciones dirigidas (e.g., aumento de la medicación), podría mejorar de manera significativa la calidad de vida de los pacientes y sus cuidadores, el pronóstico a largo plazo, y disminuir la carga a los sistemas de salud resultante de las hospitalizaciones. Desafortunadamente, en la actualidad no existen técnicas que permitan anticipar la ocurrencia de los episodios de descompensación antes de la aparición de manifestaciones clínicas evidentes (e.g., incremento en la frecuencia de ocurrencia de las alucinaciones verbales).

Por otra parte, el electroencefalograma (EEG) es una de las pocas técnicas capaces de medir, de manera no invasiva, la actividad eléctrica de las neuronas con suficiente resolución temporal (milisegundos) como para capturar el procesamiento y transmisión de información

(i.e., la dinámica) del cerebro a la escala temporal en que ocurre. Si bien la resonancia magnética funcional (fMR) posee una resolución espacial muy superior a la del EEG, es una medida indirecta de la actividad cerebral (cambio en los niveles de oxigenación sanguínea) cuya resolución temporal (~ 1 segundo) es insuficiente para capturar la dinámica rápida de los procesos cerebrales [3]. El bajo costo, carácter no invasivo, disponibilidad y alta resolución temporal hacen del EEG una técnica ideal para investigar los cambios en la dinámica cerebral de los pacientes con EQZ subyacentes a los episodios de descompensación.

Es por lo anterior, que han surgido diversas investigaciones tomando registros de EEG a sujetos con EQZ, y numerosos estudios han reportado particularidades en las dinámicas temporales de EEG en estos pacientes, comparados con sujetos control [4, 5]. A pesar de esto, no se ha llegado a un consenso aún con respecto a las anomalías específicas que presenta un paciente EQZ con respecto a los sujetos control. Si bien se ha observado en la EQZ la atenuación en los registros EEG de la banda gamma, además de patrones disruptivos en la banda delta [6, 7], no se han establecido medidas en EEG que permitan estimar el nivel de descompensación, en un momento dado, en pacientes EQZ.

No obstante lo anterior, se ha mostrado un incremento en el ruido neural (i.e. la aleatoriedad de la actividad cerebral) en el EEG de pacientes EQZ [8, 9], y más importante aún, estos cambios son mayores durante los períodos de exacerbación sintomática [10]. Este hallazgo es uno de los que motivan a pensar que el EEG podría informar de rasgos de la actividad cerebral de los pacientes, que puedan ser informativos con respecto al estado de compensación/descompensación clínica de estos.

Por otra parte, la EQZ se caracteriza por su heterogeneidad, tanto clínica como biológica, para cada paciente, debido a la gran cantidad de síntomas diferentes y variadas causas subyacentes que son aún desconocidas. Sin embargo, los estudios *case-control* suelen ignorar tal heterogeneidad al centrarse en el paciente promedio, lo cual podría ser la principal razón de la falta de biomarcadores robustos que indiquen la respuesta y resultado del individuo al tratamiento [11]. Este problema motiva que el proceso de monitoreo de los pacientes sea complementado con información empírica acerca de los niveles de sintomatología manifestados, como una propiedad más de la data recolectada, lo que se conoce como metadatos [12]. En otras palabras, los metadatos se definen como información estructurada que describe, explica, ubica o de otra manera facilita el uso o manejo de algún recurso informativo [13].

En esta tesis se desea desarrollar e implementar un modelo basado en redes neuronales que reciba una señal EEG y extraiga información relevante, y así, discrimine acerca del estado clínico del paciente. Además, se refina esta información por medio de la información otorgada por los metadatos asociados.

1.2. Hipótesis

Las hipótesis de esta tesis son:

- Se puede discriminar el nivel de compensación de un paciente EQZ a partir de la observación de un registro EEG y sus registros previos mediante aprendizaje de máquinas.
- Existen biomarcadores en registros EEG que caractericen los estados de compensación

y descompensación clínica en un mismo sujeto EQZ.

- Existe una correlación importante entre el comportamiento de descompensación de un paciente EQZ y sus evaluaciones clínicas asociadas.

1.3. Objetivo general

El objetivo general es detectar el nivel de descompensación clínica en pacientes EQZ por medio de secuencias de registros EEG, aplicando métodos de procesamiento de señales, teoría de la información y *machine learning* (ML). El enfoque, idealmente, debe ser capaz de identificar signos tempranos de descompensación en el EEG antes de que signos conductuales de descompensación (e.g. aumento de las creencias delirantes) sean detectables por una evaluación clínica tradicional.

1.4. Objetivos específicos

Los objetivos específicos son:

- Desarrollar una *pipeline* de preprocesamiento automático para limpiar señales EEG que presenten un alto nivel de ruido.
- Desarrollar un clasificador de estado clínico (estabilidad y descompensación) a partir de los registros EEG de pacientes EQZ, utilizando técnicas de *machine learning*, procesamiento de señales y teoría de la información.
- Identificar biomarcadores de la EQZ en el EEG.
- Evaluar y comparar técnicas del *deep learning* con técnicas tradicionales de *machine learning*, y en caso de existir, del estado del arte.
- Evaluar la posibilidad de aprovechar parámetros obtenidos en otras tareas relacionadas con la esquizofrenia.

1.5. Organización del documento

A continuación se describe cómo está organizado el documento. En el Capítulo 2 se describen los antecedentes necesarios para introducir al lector en el contexto de esta tesis, así como también se mencionan los hallazgos relevantes relacionados. Esto viene acompañado de una muestra teórica de los contenidos que abarcan la metodología del trabajo, y todo lo necesario para poder comprender a cabalidad el proceso.

En el Capítulo 3 se plantea el problema a resolver y las soluciones propuestas. Además se presentan las bases de datos sobre las que se trabaja y sus principales características. En el resto del capítulo se explica todo lo que es la metodología y procedimientos llevados a cabo, así como también de qué manera son tratados los datos y cómo se evalúa el desempeño de los métodos propuestos.

En el Capítulo 4 se muestran los principales resultados obtenidos, acompañados de texto explicativo. En el Capítulo 5 se discuten e interpretan estos resultados.

Finalmente, en el Capítulo 6 se recapitula todo lo visto en el trabajo de tesis, interpretando y autoevaluando los resultados obtenidos; y además, se introducen las proyecciones de trabajo futuro.

2. Marco teórico

2.1. Esquizofrenia

La esquizofrenia (EQZ) es un complejo y heterogéneo síndrome cognitivo y del comportamiento, el cual pareciera deber su origen a una disrupción del desarrollo cerebral por causas genéticas y/o ambientales [14].

Las personas con esquizofrenia suelen observarse como si hubiesen perdido el contacto con la realidad, pues de manera general los síntomas provocan una anormal interpretación de la realidad. Entre los variados tipos de síntomas de esquizofrenia -cognitivos, neurológicos, conductuales y emocionales- estos se categorizan entre 2 subgrupos, los cuales se describen a continuación:

- **Síntomas positivos:** Cuando algo se agrega a la experiencia de una persona, lo cual no es normalmente experimentado por otras personas. Se incluyen: alucinaciones, delirios, desorden del lenguaje, comportamiento desorganizado o catatónico.
- **Síntomas negativos:** Cuando existe ausencia o subdesarrollo de comportamientos que las personas deberían experimentar normalmente. Se incluyen: aplanamiento afectivo, disminución de la motivación, incapacidad para proponer y lograr metas, reacciones lentas, disminución del habla, anhedonia (pérdida de habilidad para sentir placer o felicidad), apatía (pérdida del interés por las personas y los eventos).

Todos los pacientes poseen alguna combinación de los anteriores síntomas, pero ninguno los tiene todos [15]. Cualquiera que sea la combinación, estos síntomas perturban en gran medida la vida de los pacientes. Sin embargo, con tratamiento la mayoría de los síntomas disminuyen y la probabilidad de recurrencia de estos se hace menor, pero no desaparece.

Los tratamientos incluyen medicación antipsicótica, terapia psicológica, apoyo social y rehabilitación, además del tratamiento de los efectos adversos de la medicación.

2.2. Electroencefalograma (EEG) y su aplicación en EQZ

El electroencefalograma (EEG) es una medida muy singular acerca de la función eléctrica del cerebro. Este expone visualmente la diferencia de voltaje entre un sitio del cerebro y una referencia, lo cual se registra temporalmente (ver fig. 2.1). Los EEG pueden ser extracraneales o intracraneales, donde en el primer caso se cuenta con una amplia inspección de la actividad

cerebral a lo largo de ambos hemisferios cerebrales, mientras que el segunda proporciona un registro enfocado directamente desde el cerebro al implantar electrodos quirúrgicamente en regiones específicas de este [16].

Este trabajo se enfoca en los EEG extracraneales, los que, a pesar de poseer limitada resolución espacial, continúan siendo una herramienta importante en la investigación y diagnóstico de enfermedades neurológicas, esto debido a su amplia disponibilidad, su naturaleza no invasiva, su bajo costo y su alta resolución temporal (del orden de los milisegundos).

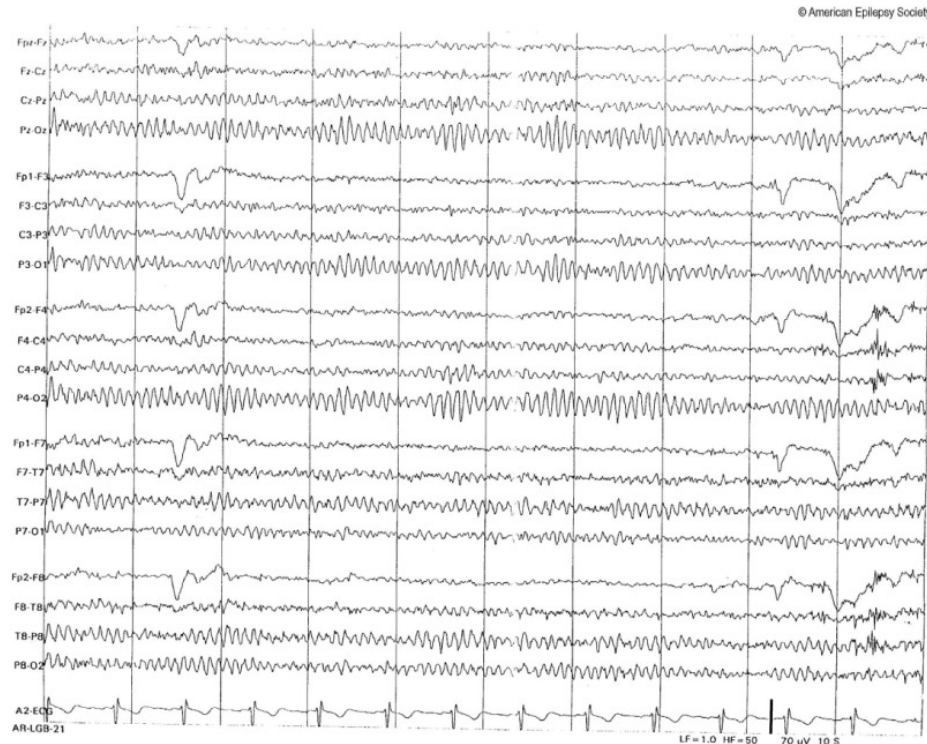


Figura 2.1: Muestra de EEG normal de 20 canales acompañado de un canal de electrocardiograma; registrado durante *wakefulness* en una mujer de 24 años. Obtenido de [17].

El EEG consta de varios canales de registro y como mínimo se aceptan 8 canales. En la práctica clínica diaria, equipos entre los 8, 16, 32 y hasta 64 canales (ver fig. 2.2) son los de uso más frecuente[18].

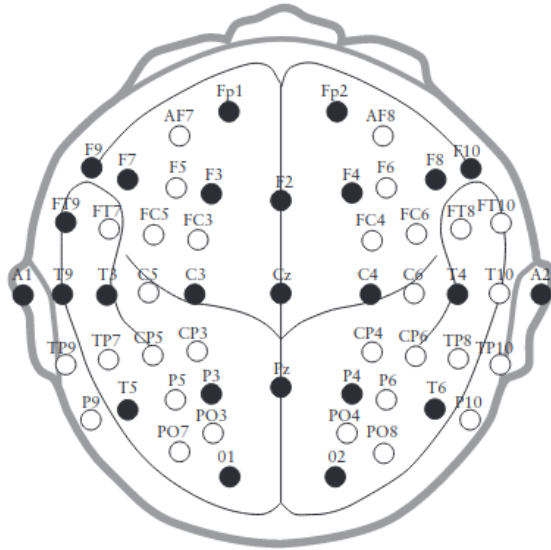


Figura 2.2: Ejemplo de ubicación de electrodos usando ya sea un sistema 10-20 (círculos negros), o un sistema combinacional modificado (círculos blancos + círculos negros)[16].

Por otra parte, en el EEG la distribución topográfica y frecuencia de las señales son fundamentales para identificar los distintos ritmos cerebrales. Las frecuencias generadas por actividad cerebral en un EEG oscilan entre 0.01 Hz a 100 Hz, las que se pueden dividir en cinco ritmos de frecuencia. Las características de estos ritmos de frecuencia y algunos hallazgos reportados en estudios con EQZ se describen a continuación:

- **Delta (<4 Hz)**: No suele ser registrable en la persona despierta con ojos cerrados, con excepción de la actividad lenta posterior de la juventud, la cual se mezcla con actividad alfa y se desprende de su fase ascendente. Alcanza amplitud moderada y presenta una frecuencia alrededor de los 3 Hz. Las personas EQZ han mostrado patrones EEG disruptivos, lo cual se asocia a la reducción de ondas delta durante el sueño y síntomas negativos asociados a la EQZ. Durante el sueño de ondas lentas (etapas III y IV), las personas EQZ han mostrado reducida actividad delta, aunque también se ha evidenciado un incremento de estas ondas durante las horas despierto en las formas más severas de EQZ [6]. Un reciente estudio, ha mostrado que la dominancia de las ondas delta en las áreas derecho-frontal y central (vista en individuos sanos), no se manifiesta en pacientes EQZ. Además, la correlación negativa entre la actividad delta y la edad tampoco se observa en personas EQZ[19].
- **Theta (4-8 Hz)**: De bajo a moderado voltaje y no supera los 100 μV . Distribuida en todas las áreas con más amplitud en la región fronto-centro-temporal bilateral. La transición vigilia-sueño se caracteriza por disminución de actividad alfa, en ocasiones incremento de actividad rápida beta y más constantemente la aparición de actividad lenta theta en todas las áreas.
- **Alfa(8-12 Hz)**: Actividad dominante en personas despiertas, recostadas y con los ojos cerrados. Mayor amplitud en región occipital, pero puede observarse en toda la región temporo-parieto-occipital de manera bilateral. Generalmente se atenúa al abrir los ojos o ante estímulos auditivos. Suele presentar amplitud variable. En promedio, y según edad, oscila entre 50 μV y 120 μV .

- **Beta (12-30 Hz):** De bajo voltaje ($<50 \mu V$ y rara vez es mayor a $15 \mu V$). Predominante en áreas frontales y frontales superiores. Suele ser opacada por artefactos musculares.
- **Gamma (30-100 Hz):** Correlacionada con actividad cerebral de gran escala y actividad cognitiva, y su amplitud puede ser aumentada por medio de meditación o neuroestimulación. La actividad gamma se ha visto alterada en desórdenes del ánimo y cognitivos (alzheimer, epilepsia, esquizofrenia, entre otros). La validez de su detección por EEG suele ser cuestionada por su superposición con la actividad muscular, es por ello que debe ser tratada con cuidado para reducir la presencia de artefactos. Se ha reportado la disminución de su actividad en pacientes EQZ. Específicamente, su amplitud se ve reducida, así como la sincronía de diferentes regiones cerebrales involucradas en tareas de percepción o memoria continua de reconocimiento, donde los pacientes EQZ se desempeñan peor. Se cree que la disfunción gamma en la EQZ reside en las interneuronas GABA-érgicas¹, involucradas en conocidas redes de generación rítmica de ondas cerebrales. Se ha visto que el tratamiento antipsicótico no restaura la sincronía gamma a niveles normales [7].

Como se puede apreciar, son abundantes los estudios que han reportado particularidades en las dinámicas temporales de EEG en los pacientes con EQZ, comparado con los sujetos control [4, 5, 21]. A pesar de esto, y de lo reportado en estudios respecto al comportamiento en las bandas gamma y delta [6, 7], no se ha llegado a un consenso aún con respecto a las anormalidades específicas que presenta un paciente EQZ, con respecto a los sujetos control en los registros EEG. Menos aún, se han encontrado hallazgos por banda con respecto al nivel de descompensación de cada paciente, en la bibliografía revisada.

A pesar de lo anterior, y como se mencionó en la sección 1.1, si se ha reportado un incremento en el ruido neural (i.e. la aleatoriedad de la actividad cerebral) en el EEG de pacientes con EQZ [8, 9] y, es especialmente relevante, que estos cambios se incrementan aun más durante los períodos de exacerbación sintomática [10]. Esto sugiere que el EEG podría ser capaz de capturar rasgos de la actividad cerebral de los pacientes EQZ que sean indicativos del riesgo de descompensación clínica.

Cabe destacar que en la bibliografía, generalmente, los análisis sobre EEG se realizan sobre segmentos de la señal, en vez de sobre la señal completa. Esto debido a los que EEG suelen ser no-estacionarios y muchas técnicas de análisis están teóricamente definidas solo para señales estacionarias. La idea es definir una longitud de segmentos tal que sea lo suficientemente larga para que el lapso sea informativo, pero lo suficientemente corta para que la señal pueda considerarse estacionaria [22].

2.2.1. Artefactos y ruido en el EEG

En procesamiento de señales, el ruido se define como toda señal no deseada que se combina con la señal útil que se desea captar. En el contexto del EEG, la señal útil o que se desea capturar corresponde a toda señal generada puramente por la actividad cerebral. Lamentablemente, todos los registros de EEG poseen ruido, también llamado artefactos, los cuales contaminan la calidad de la señal. Quien interpreta una señal EEG siempre debe estar

¹ Neuronas inhibitorias del sistema nervioso, las cuales juegan un rol vital en los circuitos neuronales y su actividad. Su nombre se debe a que liberan el neurotransmisor ácido gamma-aminobutírico (GABA)[20].

consciente de la posibilidad de que una forma de onda sea un artefacto [17].

Para lograr remover eficientemente los artefactos y el ruido se requiere poseer un conocimiento integral de los diferentes tipos de artefactos. Los artefactos se categorizan según sus fuentes de origen, principalmente se categorizan en:

- **Intrínsecos o fisiológicos/biológicos:** Incluyen artefactos generados por actividad cardíaca, pulso sanguíneo, respiración, sudor, movimientos de lengua, movimiento de ojos (*blinks* (parpadeos), *saccades* (movimiento lateral de los ojos)), actividad muscular y movimiento. Ver figura 2.3.
- **Extrínsecos o no-fisiológicos:** Incluyen artefactos generados por fenómenos eléctricos, por los aparatos o por el ruido presente en el ambiente donde se obtiene el registro.

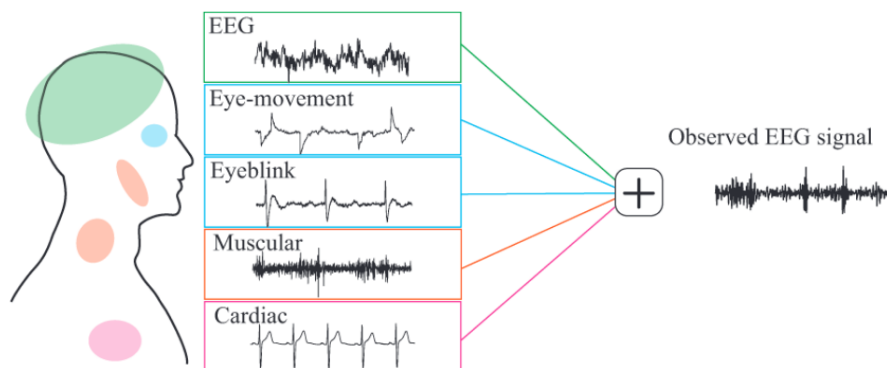


Figura 2.3: Ejemplo de una señal EEG observada que incluye artefactos biológicos. Figura obtenida de [23].

Muchos artefactos ambientales pueden eliminarse con un simple filtrado, debido a la inconsistencia de sus frecuencias con las señales cerebrales. Por otro lado, un procedimiento y planificación adecuada permitiría reducir el error experimental adecuadamente. Sin embargo, los artefactos de origen fisiológicos son más difíciles de remover, por lo que se requiere de algoritmos específicos para ellos [24].

2.3. Complejidad de Lempel-Ziv

La complejidad de Lempel-Ziv (LZC, por sus siglas en inglés) es una medida no lineal, usada en teoría de la información. Esta cuantifica el nivel de aleatoriedad o ruido de una señal [25]. El algoritmo para medir la LZC, en el contexto de los registros EEG, se resume a continuación. Primero, la información numérica del registro EEG en cada instante de tiempo es transformada a información simbólica o binaria. Para llevar a cabo esta transformación es necesario definir previamente un criterio para ello. Tradicionalmente, se utiliza algún criterio basado en un umbral fijo. Sea S la señal EEG, entonces su correspondiente señal binaria, S_{bin} , está dada por

$$S_{bin}(x) = \begin{cases} \text{si } S(x) \geq \mu & 1 \\ \text{si } S(x) < \mu & 0 \end{cases}, \quad (2.1)$$

donde μ corresponderá al umbral en cuestión, el cual comúnmente corresponde a la mediana de la señal $S(x)$ completa.

Luego de transformar la secuencia, esta será leída de izquierda a derecha. La LZC corresponde al número de diferentes patrones o *substrings* (“palabras”) presentes en la secuencia simbólica. En la fig. 2.4 se representa el proceso recién descrito. Finalmente, el número de “palabras” obtenido suele ser normalizado para evitar que este dependa del largo de la señal analizada, para lo cual existen diferentes métodos.

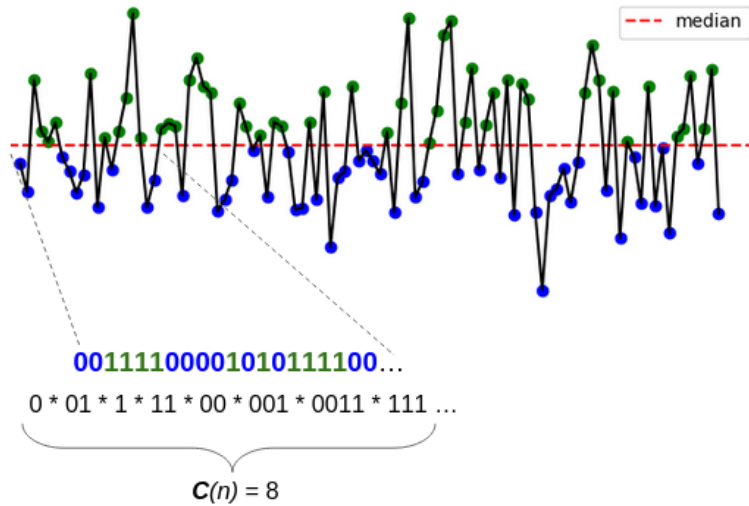


Figura 2.4: Ejemplo gráfico de cálculo de LZC, para segmento de una señal.

Para contar el número de *substrings* en la secuencia simbólica existe más de un método, lo importante es ser consistente con el método elegido al usar la métrica sobre diferentes señales. En el marco de esta tesis, el algoritmo [26] utilizado para computar el número de *substrings* de una secuencia binaria consiste en la codificación de segmentos futuros de la secuencia, contenidos en un *look-ahead buffer* (LA), esto a partir de un *search buffer* (SB) que contiene una porción de la secuencia que ya fue previamente codificada. Después de finalizar el algoritmo, y con el objetivo de normalizar el resultado, el número de *substrings* resultante es ponderado por $\log_2(n)/n$, donde n corresponde al largo de la secuencia completa, esto reduce la influencia del largo de la frecuencia en el valor de LZC obtenido. Dado $x_{seq} \in \{0, 1\}^n$ una secuencia binaria, el algoritmo 1 describe cómo computar su LZC.

Algoritmo 1: Método para computar el número de *substrings* de una secuencia binaria y obtener valor de LZC.

```

u ← 0 ; #Puntero-1
v ← 1 ; #Largo del componente actual para el puntero actual
w ← 1 ; #Largo del prefijo actual
vmax ← 1 ; #Almacena v del patrón más largo en el "look-ahead buffer" (LA) que ha calzado
en alguna parte del "search buffer" (SB)
C ← 1 ; #Complejidad
while True do
  if xseq[u + v - 1] = xseq[w + v - 1] then
    #Se puede reconstruir xseq[w+v-1] a partir de xseq[u+v-1]
    v = v + 1;
    if w + v ≥ n then
      #Se alcanza final de secuencia
      C = C + 1;
      break;
    end
  else
    if v > vmax then
      vmax = v;
    end
    u = u + 1 ; #Bit no coincide al buscar ocurrencia previa del patrón
    if u = w then
      #Detener búsqueda cuando u calza con primer bit de LA
      C = C + 1;
      w = w + vmax ; #Mover inicio de LA al final de nuevo patrón calzado
      if w ≥ n then
        #Detenerse si LA sobrepasa el largo de la secuencia
        break;
      else
        u = 0 ; #Reiniciar índice de búsqueda a inicio del SB (inicio de la secuencia).
        v = 1 ; #Reiniciar índice de coincidencia de patrón
        vmax = 1 ; #Reiniciar largo máximo del patrón calzado.
      end
    else
      v = 1 ; #Finaliza la coincidencia de patrón en el SB, y se reinicia contador de largo
de patrón coincidente.
    end
  end
end
end
Cnorm = (C * log2(n))/n ; #Normalización
return Cnorm

```

2.4. Aprendizaje de máquinas

El aprendizaje de máquinas (*machine learning* en inglés) es una rama, en constante desarrollo, de algoritmos computacionales. Está diseñado para aprender utilizando información

del ambiente actual, y así, ser capaz de generalizar en tareas no vistas previamente, emulando de cierta forma la inteligencia humana[27]. Dicho de otra manera, posee la cualidad de aprender de los datos disponibles y de esta manera mejorar el rendimiento en algún conjunto de tareas[28]. Es una rama amplia y diversa, la cual se ha aplicado de manera exitosa en una gran variedad de campos, los cuales van desde el reconocimiento de patrones, la visión computacional, ingeniería de naves espaciales, finanzas, entretenimiento, y biología computacional, entre otros.

Más específicamente, un algoritmo de aprendizaje de máquinas es un proceso computacional que utiliza datos de entrada (*inputs*) para lograr alguna tarea deseada, pero sin ser literalmente programado (i.e., “*hard-coded*”) para producir un determinado resultado. El algoritmo entonces, automáticamente altera su arquitectura para adaptarse por medio de la repetición (i.e. experiencia) de tal manera que se mejora más y más en la realización de la tarea que se desea cumplir.

A continuación se explican los conceptos de aprendizaje de máquinas más importantes en lo que respecta a este trabajo.

2.4.1. Redes neuronales artificiales

Las redes neuronales son uno de los principales modelos utilizados en el aprendizaje de máquinas y conforman la base del aprendizaje profundo. Una red neuronal corresponde a un ensamblado interconectado de simples elementos de procesamiento, llamados *unidades* o *nodos*, cuya función es inspirada en las neuronas biológicas. La habilidad de procesamiento de la red es almacenada en las fuerzas de conexión entre unidades, obtenidas por un proceso de adaptación, o *aprendizaje*, sobre un conjunto de patrones de entrenamiento [29].

Una *unidad*, es decir, una sola neurona artificial, matemáticamente está dada por

$$z = \phi\left(b + \sum_{j=1}^d w_j x_j\right), \quad (2.2)$$

donde $x_j \in \mathbb{R}^d$ es la entrada; $w_j \in \mathbb{R}$ y $b \in \mathbb{R}$ son los parámetros ajustables (*entrenables*) de la neurona llamado pesos y sesgo, respectivamente; y ϕ es una función no lineal, llamada función de activación. Esta función no lineal se encarga de determinar el nivel de *activación* de la neurona frente a una cierta entrada, el cual en su forma simplificada corresponde a una respuesta limitada a dos estados, es decir, se activa la neurona o se inhibe. Funciones tradicionales que se usan para esto corresponden a las funciones sigmoide, tangente hiperbólica, ReLU (*Rectified Linear Unit*), GELU (*Gaussian Error Linear Unit*), entre otras; para más detalles revisar la sección 2.4.6.

Continuando con la idea, un grupo de m neuronas pueden procesar la misma entrada en paralelo para formar una capa neuronal de ancho o dimensión m y una salida $z \in \mathbb{R}^m$ dada

por

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} = \begin{pmatrix} \phi(b + \sum_{j=1}^d w_{1j}x_j) \\ \phi(b + \sum_{j=1}^d w_{2j}x_j) \\ \vdots \\ \phi(b + \sum_{j=1}^d w_{mj}x_j) \end{pmatrix} = \phi(\mathbf{b} + \mathbf{W} \mathbf{x}), \quad (2.3)$$

en donde ϕ se aplica componente a componente, y $b \in \mathbb{R}^m$ y $W \in \mathbb{R}^{m \times d}$ son el vector de sesgos y la matriz de pesos de la capa, respectivamente [30]. Esta corresponde a la configuración tradicional de una capa neuronal, pero actualmente, existen otros tipos de configuraciones posibles. De esta manera, se puede definir una red neuronal como un conjunto de capas neuronales apiladas verticalmente, donde la salida de cada capa corresponde a la entrada de la capa siguiente y así.

Debido a que existe una correspondencia directa entre la configuración de una red y su función matemática, es posible desarrollar redes más generales considerando configuraciones de red más complejas. Sin embargo, estas deben limitarse a ser una arquitectura *feed-forward*, en otras palabras a una que no posea ciclos dirigidos cerrados, esto para asegurar que las salidas de la red sean funciones determinísticas de las entradas [31]. También se debe tener la consideración de que la capa de salida de la red debe poseer una dimensión y activación compatible con la salida deseada. No obstante, en general las redes neuronales poseen una gran flexibilidad en cuanto al diseño de su configuración, por lo cual han surgido diversas variaciones de estas, usándose en ellas diferentes elementos, tales como *skip-connections*, capas bidireccionales, o capas tipo *sparse*, por ejemplo.

Las propiedades de aproximación de las redes *feed-forward* han sido ampliamente estudiadas y se ha visto que son muy generales. Se ha demostrado, incluso, que las redes neuronales son *aproximadores universales* [32–34]. Esto quiere decir que, por ejemplo, una red de dos capas con salidas lineales es capaz de aproximar uniformemente cualquier función continua, en un dominio de entrada compacto, a un nivel de precisión arbitrario si es que la red utiliza un número suficientemente grande de neuronas. A pesar de la gran capacidad demostrada de esta afirmación, el problema clave recae en cómo encontrar la configuración y los valores adecuados para cada parámetro dada una cierta tarea y un cierto conjunto de datos de entrenamiento a considerar.

2.4.2. Aprendizaje supervisado

Como ya se comentó, el aprendizaje de máquinas permite que un sistema o modelo adquiera conocimiento sobre algo de manera automatizada, y de esta forma, se mejore progresivamente en la realización de una cierta tarea. Esto lo logra gracias a la experiencia, donde no se requiere ser programado explícitamente para ello. Para conseguir lo anterior existen diferentes algoritmos de aprendizaje, los cuales difieren en función del paradigma de aprendizaje aplicado. Uno de estos paradigmas, corresponde al aprendizaje supervisado.

El aprendizaje supervisado es una técnica usada para ajustar, o “entrenar”, una función (modelo) a partir de un conjunto de datos de entrenamiento. A diferencia de otro tipo aprendizajes, en este caso los datos de entrenamiento corresponden a muestras de vectores de entrada (*input*) junto a sus correspondientes vectores objetivo (*target*) [31].

Aquellos casos donde, dado un número finito de categorías discretas, el objetivo es asignar cada vector *input* a una de estas categorías, es decir, un *target* discreto, recibe el nombre de problema de clasificación. Por otro lado, se le llama problema de regresión cuando el *target* deseado consiste en una o más variables continuas. El aprendizaje tiene el fin de crear una función capaz de predecir el *target*, por medio de su salida (*output*), correspondiente a cualquier objeto *input* válido después de haber visto una serie de ejemplos, que corresponderán a los datos de entrenamiento. Si el *output* de la función corresponde a una o más variables continuas, se le llamará un problema de regresión; en cambio, si el *output* corresponde a una categoría de un número finito de categorías discretas (clases), recibirá el nombre de problema de clasificación.

En este tipo de aprendizaje el objetivo es que la función aprenda a generalizar sobre diferentes tipos de datos, no que memorice los datos con los que se entrena. Es por lo anterior que los conjuntos de datos disponibles suelen dividirse en:

- Conjunto de entrenamiento: Son los datos que se usan para entrenar el modelo.
- Conjunto de validación: Son los datos usados para evaluar el desempeño del modelo de manera frecuente. Su función principal es evitar el sobreajuste y, además, afinar los hiperparámetros del modelo. La idea es ver los resultados del modelo en el conjunto de validación para mejorar los hiperparámetros, entonces el modelo generalmente ve estos datos, pero nunca aprende de ellos directamente. De todas maneras, no necesariamente debe existir un conjunto de validación.
- Conjunto de *test*: Son los datos usados para evaluar sin sesgos el desempeño final del modelo. Estos datos solamente se deben usar cuando el modelo está completamente entrenado y con sus hiperparámetros ajustados.

2.4.3. Entrenamiento por retropropagación del error

Una red neuronal representa una familia de posibles funciones f_θ parametrizada por $\theta \in \Theta$ (pesos y sesgos). En el aprendizaje supervisado, dado un *input*, x , y su respectivo *target*, y , se busca con el entrenamiento aproximar un óptimo θ^* , tal que $\hat{y} = f_{\theta^*}(x)$ se encuentre a una mínima distancia de y . Para encontrar aquella aproximación, se entrena por iteraciones, donde en cada iteración se ajustan los parámetros con el criterio de optimizar la esperanza de una cierta función de pérdida diferenciable $L(y, \hat{y})$, cuyo mínimo es alcanzado cuando $\hat{y} = y$. La elección de esta función de pérdida dependerá del tipo de problema a resolver.

En problemas de clasificación de C clases, se tendrá que $y \in [0, 1]^C$ representará la distribución de probabilidad sobre dichas clases, y en estos casos se suele utilizar la entropía cruzada como función de pérdida, es decir, se busca minimizar la entropía cruzada entre la salida de la red y la salida deseada (*target*), dada por

$$L(y, \hat{y}) = - \sum_{c=1}^C y_c \log(\hat{y}_c). \quad (2.4)$$

Si el conjunto de entrenamiento posee N muestras $(x_i, y_i)_{i=1}^N$, entonces, donde $\mathbb{E}[L(\theta)]$ es la esperanza sobre $L(y, \hat{y})$, se tiene que el entrenamiento busca encontrar la siguiente solución

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}[L(\theta)] = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N L(y^i, f_{\theta}(x^i)). \quad (2.5)$$

El hecho de que f_{θ} y L sean diferenciables permite utilizar el método de gradiente descendente, el cual funciona de manera iterativa. Sea θ_0 la inicialización y α la tasa de aprendizaje, las iteraciones están dadas por

$$\theta_{k=0} = \theta_0, \quad \theta_k = \theta_{k-1} - \alpha \Delta_{\theta} E[L(\theta_{k-1})], \quad (2.6)$$

donde el valor del gradiente puede reescribirse como

$$\Delta_{\theta} E[L(\theta_{k-1})] = \frac{1}{N} \sum_{i=1}^N \Delta_{\theta} L(y^i, \hat{y}^i). \quad (2.7)$$

Calcular estas operaciones puede implicar un alto costo computacional, por lo que se opta por estimar de forma insesgada el gradiente $\Delta_{\theta} E[L(\theta_{k-1})]$ en cada iteración considerando un lote (*batch*) aleatorio de $M \ll N$ muestras de entrenamiento. Debido a la naturaleza estocástica de la estimación recién descrita, este algoritmo recibe el nombre de gradiente descendente estocástico (SGD, por sus siglas en inglés). Especialmente en problemas de optimización de alta dimensión, esto reduce considerablemente la alta carga computacional requerida, logrando iteraciones más rápidas a cambio de una tasa de convergencia menor [35].

Se ha visto que cuando la tasa de aprendizaje α decrece a un ritmo apropiado, y sujeto a ciertas suposiciones fáciles de conseguir, el SGD converge casi seguramente a un mínimo global cuando la función objetivo es convexa, o pseudoconvexa, y de otra manera, converge hacia un mínimo local [36].

Cabe mencionar, que para calcular el gradiente $\Delta_{\theta} L(y^i, \hat{y}^i)$ para cada iteración, se utiliza el algoritmo de retropropagación del error (*error backpropagation* en inglés). Este funciona aprovechando la regla de la cadena, pues se calcula el gradiente del error de la función de pérdida con respecto a cada peso de la red, desde la salida de la red a la entrada de ésta, considerando que la red es una gran composición de funciones. De esta manera se obtiene el gradiente con un costo computacional similar a la evaluación de la red hacia adelante.

2.4.4. Aprendizaje profundo

El aprendizaje profundo (en inglés *deep learning*) es un subconjunto de métodos dentro del área de aprendizaje de máquinas, y el cual debe su nombre, “profundo”, al uso de un elevado número de capas neuronales para extraer progresivamente características (*features*) de alto nivel, a partir de una entrada en crudo (*raw*). Este subconjunto de métodos ha obtenido gran popularidad en los últimos años debido a sus notables resultados en el procesamiento de imágenes, video, audio y texto [37], el cual ha sido adoptado progresivamente por otras diversas áreas y, además, ha renovado su popularidad debido a los grandes avances en *hardware*, haciendo más sencilla y eficiente su implementación [38].

Al tratarse de un subgrupo del aprendizaje de máquinas, el aprendizaje profundo de igual forma funciona recibiendo muestras de entrada (*inputs*) de un conjunto de datos y, a partir de aquello, se entrena ajustando sus parámetros para lograr mejorar en la tarea que se le

desea asignar. Sin embargo, lo que hace particular al aprendizaje profundo es que, a la vez que ajusta su modelo para resolver la tarea asignada, simultáneamente aprende la mejor manera de representar los *inputs* para cumplir aquella tarea. En otras palabras, no es necesario entregarle al modelo las características (*features*) de los datos que, a priori, los expertos creen que puedan ser de importancia para cumplir la tarea, sino que el mismo modelo se encarga de buscar qué aspectos son de mayor importancia en los datos *raw* o mínimamente preprocesados y con alta dimensionalidad.

La base del aprendizaje profundo está fuertemente constituida por el uso de redes neuronales, y es que se necesita de un elevado número de capas neuronales para lograr obtener una representación eficiente de los datos y, a su vez, realizar la tarea requerida. En otras palabras, el número de parámetros a ajustar se incrementa considerablemente, y por ende, se suele requerir de una mayor cantidad de datos para ello también [39]. Por lo anterior, los entrenamientos con aprendizaje profundo requieren de una elevada capacidad de cómputo en comparación a otras técnicas. Los conceptos más importantes, en lo que respecta a esta tesis, se describen a continuación.

2.4.5. Tipos de capas neuronales

A continuación se describen distintos tipos de capas que tienen relación con este trabajo de tesis:

- **Fully connected:** Ya descrita en la ec. (2.2). Capa donde todas las entradas están conectadas a cada neurona de la capa. En los modelos más populares de redes neuronales, las últimas capas suelen ser capas *fully connected* donde se compila toda la información extraída por las capas anteriores para formar la salida final. Una capa *fully connected* de m neuronas transforma una entrada $z_{in} \in \mathbb{R}^m$ en una salida $z_{out} \in \mathbb{R}^n$, siguiendo la ecuación $z_{out} = Wz_{in} + b$, con $W \in \mathbb{R}^{n \times m}$ la matriz de pesos y $b \in \mathbb{R}^n$ el vector de sesgos.
- **Capa convolucional:** Mediante una operación de convolución se aplica un filtro o kernel a cada entrada de la capa; en otras palabras las neuronas de la capa se agrupan en filtros, y cada filtro contiene neuronas que comparten parámetros y se conectan a una vecindad de la entrada. Así, cada filtro implementa una convolución con la entrada a lo largo del o los ejes. Sus salidas suelen recibir el nombre de *feature maps*. El cálculo de estas salidas, similar a las capas *fully connected*, está dado por

$$z_{out} = W * z_{in} + b, \quad (2.8)$$

donde W es el conjunto de filtros a aplicar por convolución sobre la entrada y b el vector de sesgos, que posee una componente para cada filtro. En este caso la operación $*$ corresponde a una convolución.

Sea $z_{in} \in \mathbb{R}^{l_{in} \times c}$, un tensor de 2D que representa una entrada de 1D de largo l_{in} y c canales, que se aplica a una capa convolucional 1D de m filtros, tamaño de filtro k . Entonces el tensor de entrada será transformado a otro tensor 2D $z_{out} \in \mathbb{R}^{l_{out} \times m}$, por medio de la convolución $z_{out} = W * z_{in} + b$, en donde $W \in \mathbb{R}^{k \times c \times m}$ es la matriz de pesos que agrupa los m filtros de forma (k, c) , y $b \in \mathbb{R}^m$ es el vector que agrupa el sesgo de cada filtro. Los casos borde de z_{in} se pueden rellenar con ceros, aplicando *zero-padding*,

para que $l_{out} = l_{in}$. Además, la convolución puede aplicarse con *stride* s (i.e. cada s posiciones); en el caso cuando $s = 1$, se convoluciona en todas las posiciones posibles.

- **Capa de *pooling*:** Se emplea para reducir el largo y el ancho (solo largo en el caso de 1D) de un tensor, por ende, se reduce la cantidad de memoria y el tiempo del entrenamiento. Además, esto mejora la invarianza de la capa a pequeñas traslaciones o desplazamientos. Generalmente se usan para disminuir el tamaño de los *features maps* que generan las capas convolucionales.

Las capas de *pooling* funcionan deslizando un filtro de 1D o 2D (dependiendo la dimensión de la entrada) sobre el canal del *feature map* de entrada y resumiendo las *features* que caigan dentro de la región cubierta por el filtro. Existen diferentes tipos de capas de *pooling*, dependiendo del criterio usado para resumir las *features* cubiertas por el filtro deslizando. Los principales tipos son:

- *Max pooling*: Selecciona el elemento con el valor más alto de la región del *feature map* cubierta por el filtro. Así, su salida será un *feature map* con las *features* más prominentes del *feature map* de entrada.
- *Average pooling*: Calcula el promedio de los elementos presentes en la región del *feature map* cubierta por el filtro.
- *Global pooling*: Reduce cada canal del *feature map* a un solo valor. Así, un *feature map* de dimensión $n_l \times n_c$ se reducirá a $1 \times n_c$. Podría ser, además, *global max pooling* o *global average pooling*.

2.4.6. Tipos de funciones de activación

A continuación se describen las funciones de activación más comúnmente usadas, acompañado de la figura 2.5 que las representa.

- **Sigmoide:** Toma cualquier valor real y lo transforma en un valor entre 0 y 1. Matemáticamente, se define como

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (2.9)$$

- **Tangente hiperbólica (*tanh*):** Es similar en forma a la función sigmoide. Esta toma cualquier valor y lo transforma en un valor entre -1 y 1. Matemáticamente, se define como

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2.10)$$

- ***Rectified Linear Unit* (ReLU):** Función de activación más usada hoy en día. Matemáticamente, se define como

$$\text{ReLU}(x) = \max(0, x). \quad (2.11)$$

Tanto la función como su derivada son monotónicas. Tiene la ventaja de acelerar el proceso de aprendizaje y, además, sobreponerse a algunos problemas que presentan la sigmoide y la *tanh*. Particularmente, es menos susceptible al problema de *vanishing gradient*, pero si es vulnerable a sufrir de otros problemas como el problema de las unidades saturadas o “muertas”.

- ***Gaussian Error Linear Unit (GELU)***: Función que, en teoría, multiplica su entrada por la función de densidad acumulada de una distribución Gaussiana. Dado que su cálculo es demoroso, generalmente en la práctica se utiliza una aproximación bastante certera de esta, en este caso matemáticamente, se define como

$$GELU_{aprox} = 0.5 \cdot x \cdot 1(1 + \tanh(\sqrt{2/\pi} \cdot (x + 0.044715 \cdot x^3))). \quad (2.12)$$

A diferencia de la familia de activaciones ReLU, la función GELU es no convexa y no monotónica, además es no-lineal en el dominio positivo y presenta curvatura en todos sus puntos [40]. Su curvatura en aumento y no-monotonicidad le permite aproximarse más fácilmente a funciones más complejas, en comparación a ReLU o ELU. También, al ser más suave en las cercanías de 0, y ser diferenciable en todo el rango, permite tener gradientes, aunque pequeños, en el rango negativo, lo cual ayuda con el problema que puede ocurrir con ReLU cuando un considerable número de neuronas llegan a 0 y, por ende, aquellas unidades prácticamente no aportan en nada (“*dying ReLU problem*”).

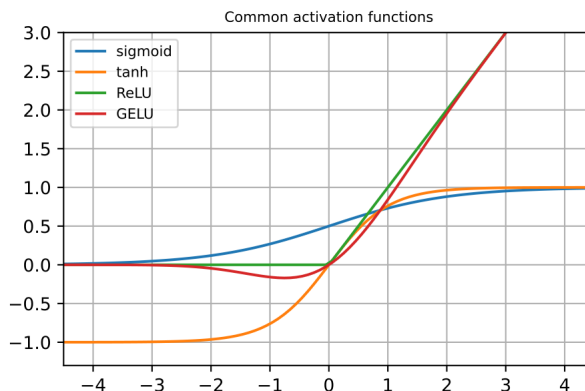


Figura 2.5: Funciones de activación más utilizadas en las arquitecturas de redes neuronales.

2.4.7. Transferencia de aprendizaje

En *deep learning* cuando se quiere usar un red o modelo para cierta tarea, es importante considerar cómo se inicializarán los parámetros de aquel modelo antes de comenzar a entrenar. Históricamente esto se ha hecho con pequeños valores aleatorios, pero en el último tiempo se han desarrollado otras heurísticas más específicas que toman en consideración tanto el tipo de función de activación que se usará como el número de *inputs* que ingresarán a la red.

La transferencia de aprendizaje, en inglés *transfer learning*, es una metodología usada en *deep learning* en la que un modelo, que ya se ha entrenado antes para cierta tarea, se utiliza como punto de inicio en un nuevo entrenamiento para realizar una tarea similar, es decir, se reutiliza tanto la arquitectura del modelo como los parámetros que se obtuvieron anteriormente, como punto de partida para la nueva tarea. En otras palabras, la idea es aprovechar el conocimiento obtenido al resolver una tarea anterior para aplicarlo en un problema diferente pero relacionado. Este último punto es importante, pues es fundamental que la nueva tarea tenga algún tipo de relación con la tarea anterior para que esto cobre sentido, porque de otra manera el conocimiento previo que se querrá usar podría confundir al modelo y resultar

ser contraproducente. El proceso de entrenar utilizando como condiciones iniciales los parámetros obtenidos previamente en otro entrenamiento con otros datos, recibe el nombre de afinamiento (*fine-tuning*).

Esta metodología posee las siguientes ventajas:

- Permite entrenar con menos datos etiquetados, pues los parámetros iniciales ya se consiguieron entrenando con una gran cantidad de datos.
- Reduce el tiempo de entrenamiento y los recursos computacionales necesarios.
- Permite aprovechar arquitecturas de modelos desarrollados por la comunidad investigadora de *deep learning*.

2.4.8. Cuidados con el gradiente en el entrenamiento

En el ajuste o entrenamiento de un modelo, el gradiente tiene un papel central, pues es quién guía la influencia de los parámetros en la tarea solicitada y, de esta manera, guía el entrenamiento. Por lo mismo, son variadas las técnicas que se han propuesto para mantener su “calidad” y evitar caer en divergencias o estancamientos del entrenamiento. Algunas serán descritas a continuación:

- **Función de activación:** La elección de la función de activación es fundamental, pues muchas veces esta puede ser la causa de problemas en el gradiente. Un ejemplo de esto, son las funciones sigmoide o tanh, las cuales resultan susceptibles al problema de desvanecimiento del gradiente. Es por lo anterior, que existen otras opciones para evitar ciertos problemas específicos, lo cual no resuelve la aparición de todos los problemas posibles. Un ejemplo de esto es la función ReLU, la cual resulta poco susceptible al desvanecimiento del gradiente, pero si resulta susceptible al problema de las neuronas “muertas”, lo cual ocurre cuando ciertas neuronas presentan una activación 0 y luego, difícilmente salen de aquél estado.
- **Inicialización de los pesos o parámetros:** Es importante tener un buen criterio de inicialización de los pesos. De otra manera, es fácil que se produzca un gradiente explosivo que haga diverger el entrenamiento, o por otro lado, un gradiente que se desvanece, el cual provoque la detención del aprendizaje.
- **Criterio de actualización:** El entrenamiento es dirigido, en palabras simples, por la tasa de aprendizaje y la dirección de descenso. Es por lo anterior que resulta fundamental este aspecto, y por lo mismo, se han propuesto variados métodos para su ajuste. Uno de los más populares es el optimizador *Adaptive moment estimation* (Adam), el cual es una extensión del optimizador SGD. Adam introduce lo que es una tasa de aprendizaje específica para cada parámetro, lo cual mejora el desempeño en problemas de gradiente disperso, y adapta estas tasas de aprendizaje basándose en el promedio exponencial de las magnitudes recientes del gradiente.
- **Gradient clipping:** Técnica sencilla pero efectiva para combatir el gradiente explosivo. Funciona acotando el gradiente a un cierto umbral (*threshold*), esto mejora el desempeño especialmente cerca de las áreas más “afiladas” de la superficie de la función de pérdida.

2.4.9. Cuidados con la generalización del modelo

Lograr un alto nivel de generalización es la principal finalidad al entrenar un modelo. Es por lo mismo que el objetivo primordial no será minimizar la función de pérdida en el conjunto de entrenamiento, sino que aprender una función lo suficientemente general, en la tarea, que minimice la esperanza de la función de pérdida bajo la distribución de probabilidad desconocida de los datos. Es por lo anterior que las métricas de desempeño se realizan sobre el conjunto de *test*. Para lograr lo anterior, la red debe ser lo suficientemente compleja para tener la capacidad de abordar la tarea pedida, pero tampoco demasiado compleja para evitar el riesgo de sobreajustarse al conjunto de entrenamiento.

Para ayudar a conseguir lo anterior, son variadas las métodos que regularizan el entrenamiento para disminuir el riesgo de sobreajuste. Los métodos relacionados con esta tesis, se describen a continuación:

- ***Data augmentation***: Existe una clara correlación entre el número de datos para entrenar, y la capacidad de generalización del modelo resultante. Es por lo anterior que el método de *data augmentation* puede ser de gran utilidad. Este consiste en aumentar artificialmente el conjunto de datos de entrenamiento. Existen variados métodos para hacer esto, desde modificar levemente los datos ya existentes para crear “nuevos”, hasta generar datos nuevos a través de un modelo de aprendizaje profundo entrenado con los datos actuales.
- **Dropout**: Consiste en “apagar” ciertas neuronas durante el entrenamiento de manera aleatoria. Esto resulta en la disminución de la dependencia sobre neuronas específicas. Además, su funcionamiento aproxima entrenar un gran número de redes neuronales de diferentes arquitecturas en forma paralela y luego promediarlas.
- **Regularización L2**: Combate el sobreajuste de un modelo forzando a que los pesos posean valores pequeños, pero sin hacerlos exactamente 0. Para esto, funciona sumando un término de regularización a la función de *loss*, el cual es multiplicado por la suma del cuadrado de todos los pesos.

2.4.10. *Random Forest*

Random Forest (RF) es un método de aprendizaje ensamblado, el cual se usa para tareas de clasificación o regresión. El método consiste en construir múltiples árboles de decisión durante el entrenamiento.

- **Árbol de decisión**: Técnica de aprendizaje supervisado para obtener conclusiones acerca de un conjunto de observaciones.

La variable *target* que se busca predecir puede ser discreta (tarea de clasificación) o continua (tarea de regresión). Un árbol de decisión está conformado por hojas, que representan los valores que puede tomar la variable *target* que se busca predecir. Esto implica que si se quiere resolver una tarea de regresión, la variable a predecir solo puede tomar un set de valores discretos; sin embargo, si el árbol es lo suficientemente complejo la cantidad de valores posibles puede ser lo suficientemente específica. Además de las hojas, un árbol posee ramas, las que representan conjunciones de características, las cuales llevan a las diferentes hojas del árbol.

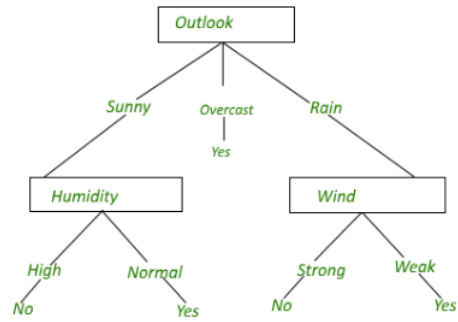


Figura 2.6: Ilustración de árbol de decisión simple para decidir si salir a jugar tennis o no. Las hojas son las 4 opciones presentes al inferior de la imagen, y corresponden a los valores que puede tomar la variable *target*. Las ramas están representadas por los recuadros.

Para tareas de clasificación, la salida de un RF será la clase seleccionada por la mayoría de los árboles de decisión que lo conforman. Si la tarea es de regresión, la salida del RF será el promedio de las predicciones de los árboles de decisión.

3. Metodología

3.1. Descripción general del problema

3.1.1. Datos

Se dispone de una colección $S = \{s_i\}_{i=1}^N$ de N sujetos con esquizofrenia; además cada sujeto tiene asociado una colección de M días $D_i = \{d_{ij}\}_{j=0}^M$. Para cada sujeto s_i , se registra una señal de EEG $x^{id_{ij}} \in \mathcal{R}^{T_{ij} \times C}$, correspondiente a cierto día d_{ij} , de T_{ij} muestras y C canales con frecuencia de muestreo f_s .

Por otra parte, asociados a cada registro $x^{id_{ij}}$ se tienen metadatos, desde los cuales, dado un cierto criterio (ver Sección 3.2.2), se obtendrá una etiqueta $y_{id_{ij}}$ (numérica o categórica) que identifique el estado clínico del sujeto en el momento de obtención del registro en cuestión. De esta forma se tiene una colección de etiquetas $Y^i = \{Y^{id_{ij}}\}_{j=1}^M$ asociada a cada sujeto, es decir una gran colección $Y = \{Y^i\}_{i=1}^N$.

3.1.2. Tarea a resolver

Por un lado, se desea realizar un análisis exploratorio de las características que puedan ser relevantes para estimar $\hat{Y}^{id_{ij}}$, que identifique el estado clínico del paciente i , a partir de un registro EEG $x^{id_{ij}}$.

Y por otra parte, se desea implementar un modelo f_θ , de parámetros θ , tal que $f_\theta(x^{id_{ij}}) \approx Y^{id_{ij}}$. Es decir, se desea que el modelo paramétrico f_θ sea capaz de realizar una estimación $\hat{Y}^{id_{ij}}$ que identifique el estado clínico del paciente i , a partir de un registro EEG $x^{id_{ij}}$ -el cual no proviene de los registros empleados para diseñar y ajustar el modelo- que aproxime la colección que se tiene originalmente $Y^i = \{Y^{id_{ij}}\}_{j=1}^M$.

3.1.3. Solución propuesta

Para llevar a cabo las tareas anteriores, se propone:

- (a) Extraer de características de manera manual, para su posterior análisis.
- (b) Aplicar modelo de aprendizaje profundo entrenado con transferencia de aprendizaje, el cual recibe la señal EEG en crudo como entrada.

Para el primer enfoque, y por simplicidad, se considerará solo un canal de cada señal EEG (i.e. $C = 1$); aunque para el segundo enfoque si se considerarán los 19 canales del cuero cabelludo (i.e. $C = 19$), pues no es necesario extraer características previo al modelo, por

ende es más directo. Por otro lado, dado al limitado número de registros EEG disponibles, y dada la naturaleza no-estacionaria de las señales, se decide utilizar segmentos de EEG para entrenar los modelos, y una frecuencia de muestreo fija para cada enfoque. Para ambos casos se usarán segmentos de predicción de largo 40 s para ajustar los modelos², y una frecuencia de muestreo fija para cada enfoque.

Por otra parte, se decidió implementar modelos para los registros de **un solo sujeto a la vez, y por ende, que el modelo solamente sea ajustado con los datos de ese sujeto**. Esto, debido a la naturaleza heterogénea de la EQZ, la gran variedad de síntomas que presenta, y las variaciones en su mecanismo según la edad de cada sujeto. Es decir, no se pretende por lo menos en esta tesis, implementar un modelo que funcione para cualquier sujeto con EQZ.

3.2. Base de datos

Se usaron registros EEG y metadatos conductual-cognitivos obtenidos en la *Clinical Neuroscience Research Unit* (CNRU) del *Connecticut Mental Health Center*; esto como parte de un estudio del departamento de Psiquiatría de la Universidad de Yale financiado por la *Brain and Behavior Research Foundation* (#23848). Los datos consisten en registros EEG en estado de reposo, en 22 pacientes EQZ con frecuentes episodios de descompensación, capturados regularmente, durante un período de entre 4 hasta 20 semanas. Los registros se acompañan de metadatos que comprenden el estado clínico (PANSS³, CGI⁴, SAPS⁵), historial médico, información acerca del consumo de sustancias, medidas cognitivas o motrices e información demográfica de cada uno.

A grosso modo, la información contenida en todos los metadatos corresponde a:

1. Datos demográficos: género, edad, años de educación, raza, etnia, preferencia lateralidad manos, alergias, fecha de consentimiento.
2. Historial médico pasado.
3. Historial psiquiátrico pasado.
4. Medicación previa.
5. Uso de drogas y alcohol a lo largo de la vida.
6. Formulario de uso de sustancias.
7. Cuestionario de uso de sustancias.
8. Escala de consumo de marihuana a lo largo de la vida.
9. NART (*National Adult Reading Test*).

² La elección de 40 s es por heurística, y por considerarse un período prudente para poder observar alteraciones sintomáticas, según los expertos, sin ser demasiado extenso.

³ *Positive and Negative Syndrome Scale*.

⁴ *Clinical Global Impression*.

⁵ *Scale for the Assessment of Positive Symptoms*.

10. Inventario de Edimburgo: evaluación de lateralidad cerebral a partir de la dominancia manual.
11. Exámenes de movimiento: AIMS (*Abnormal Involuntary Movement Scale*), BARS (*Barnes Akathisia Rating Scale*) y SAS (*Simpson-Angus Scale*).
12. PANSS (*Positive and Negative Syndrome Scale*).
13. CGI (*Clinical Global Impression*).
14. SAPS (*Scale for the Assessment of Positive Symptoms*).
15. TLFB (*Alcohol Lifetime Followback*).
16. Efectos adversos.
17. Estado cognitivo.

Las señales EEG fueron registradas a una frecuencia de muestreo $f_s = 1000\text{Hz}$ en 19 canales del cuero cabelludo ($F7, Fp1, Fp2, F8, F3, Fz, F4, C3, Cz, P8, P7, Pz, P4, T3, P3, O1, O2, C4, T4$), posicionados de acuerdo al *International 10-20 system*, además de un canal adicional ($A2$) posicionado en el lóbulo derecho que se utilizó como electrodo de referencia. Cada sujeto de la base de datos presenta entre 2 y 15 registros, donde cada registro corresponde a un día específico y posee una duración de 6 minutos aproximadamente, donde aproximadamente 3 minutos son de estado reposo con ojos abiertos y 3 minutos son de estado reposo con ojos cerrados. Se debe recordar que los registros de EEG que se disponen corresponden a señales originalmente analógicas, que son digitalizadas con el propósito de almacenarlas y analizarlas; consecuentemente, los valores de cualquiera de estas series de tiempo $x(t)$, $t = 1, \dots, T$ se asumirá que corresponden a medidas obtenidas desde una señal fisiológica análoga $x(\tau)$ en instantes de tiempo $\tau_t = t \frac{1}{f_s}$.

Para efectos de esta tesis, esta base de datos se denomina ***DecompStudy*** y, a pesar de que los sujetos en total son 22, solo se considerarán 9 en este estudio. Esto se debe a que muchos de ellos poseen solo 1 o 2 registros de EEG, puntajes PANSS/SAPS faltantes o situaciones específicas que generan un ruido excesivo en las señales EEG. Otras causas incluyen a sujetos que además padecían otras afecciones, como epilepsia, o sujetos extremadamente drogadictos, según la metadata entregada. A continuación, en la tabla 5.1 se presenta información general acerca de los 9 sujetos que son considerados en este estudio.

Tabla 3.1: Información general acerca de los sujetos participantes de *DecompStudy*.

Sujeto	Nº registros	Iteraciones CV ^a	Género	Edad	Medicación principal
SA007	9	4	M	22	Risperidona
SA010	9	3	M	34	Paliperidona
SA011	9	3	F	26	Risperidona
SA012	5	2	M	25	Risperidona
SA014	7	3	M	24	Haloperidol, Quetiapina, Olanzapina
SA025	6	3	M	37	Quetiapina, Olanzapina, Risperidona, Poliperidona
SA026	5	2	M	31	Clorpromazina
SA039	12	6	F	28	Haloperidol, (padece diabetes), (marihuana)
SA047	9	4	M	40	Ningún AP informado

^a Sigla corresponde a *cross-validation*, método que es utilizado en los entrenamientos.

3.2.1. Limpieza de las señales

Como se mencionó en la sección 2.2.1. los registros EEG en crudo suelen presentar una gran cantidad de ruido proveniente de diversas fuentes, de hecho, el set de datos *DecompStudy* resulta especialmente ruidoso, puesto que se trata de pacientes con una condición donde muchas veces les es difícil estar quietos o relajados. Por esto es necesario realizar una “limpieza” previa en las señales para poder capturar la información neuronal de interés. Esta limpieza se realiza mediante una *pipeline* de preprocesamiento.

Algunas de las técnicas de esta *pipeline* fueron motivadas por algunas recomendaciones de Makoto Miyakoshi [41], integrante del SCCN (*Swartz Center for Computational Neuroscience*), UCSD. También se contó con la colaboración de José Cortés-Briones, Científico Investigador asociado a la Universidad de Yale, quien proporcionó una red neuronal profunda, ya entrenada, la cual se utiliza en una de las etapas de este preprocesamiento. La *pipeline* que se describe a continuación, etapa por etapa, funciona de manera automática y fue implementada en MATLAB, utilizando específicamente el *toolbox* EEGLAB, el cual es descrito en más detalle en la sección 3.3.

La *pipeline* de procesamiento, encargada de “limpiar” las señales, se describe a continuación, donde para cada registro se aplica lo siguiente:

1. **Sólo ojos cerrados:** Se conserva solo el segmento donde el sujeto se encuentra en estado de reposo con ojos cerrados (aprox. 3 minutos).
2. **Filtro pasa-banda:** Filtrado pasa-banda con frecuencia de corte inferior 0.1 Hz (filtro pasa-alto) y frecuencia de corte superior 100 Hz (filtro pasa-bajo). Se usa un ancho de banda de transición de 1 Hz, lo que implica que el filtro es de orden 3300. El propósito del filtrado pasa-alto es para remover la *baseline* de la señal, es decir la componente continua de la señal. Por otro lado, el rango de frecuencia generado por actividad puramente cerebral no debería sobrepasar los 100 Hz, y es por esta razón que se utiliza el filtrado pasa-bajos, de esta forma se descarta gran parte del ruido.
3. **Filtro Notch:** Filtrado Notch (también conocido como elimina-banda) con frecuencia

corte inferior 59 Hz y frecuencia de corte superior 61 Hz. Al igual que con el filtro pasa-banda, este filtro presenta ancho de banda de transición de 1 Hz y un orden 3300. El propósito de este filtrado es eliminar el ruido de línea generado por la alimentación eléctrica de la red, la cual en Connecticut, EEUU, (desde donde se registraron las señales) posee una frecuencia de 60 Hz.

4. **Red neuronal:** La señal de EEG se ingresa a una red neuronal artificial profunda de tipo *encoder-decoder*, con su arquitectura inspirada en la red U-Net [42]. Esta red fue diseñada y entrenada para eliminar artefactos de baja frecuencia de la señal, principalmente oculares, sin comprometer mayormente el resto de la señal. En palabras simples, la red funciona procesando cada canal por separado, donde se realiza una codificación a una menor resolución obteniendo así una representación en espacio latente (etapa del *encoder*), la cual luego es procesada y luego llevada de vuelta a su dimensión original (etapa del *decoder*), entregando como salida una señal de igual dimensión que el canal de entrada, la cual correspondería al canal sin los artefactos que fueron detectados por la red.

El diseño y entrenamiento de la red neuronal fue llevado a cabo por José Cortés-Briones, Científico Investigador asociado a la Universidad de Yale. Tanto la arquitectura específica como los parámetros (pesos) de la red no están disponibles públicamente.

5. **Remoción de artefactos con `clean_rawdata()`:** Con el objetivo de retirar artefactos remanentes en la señal, se utiliza la extensión `clean_rawdata()` de EEGLAB [43]. Esta es capaz de detectar y separar desvíos (comúnmente llamados *drifts*) de baja frecuencia, segmentos aplanados (*flatline*) y canales ruidosos de las señales analizadas. Además es capaz de “reparar” ciertos segmentos de la señal por medio del método ASR (*Artifact Subspace Reconstruction*), el cual utiliza un filtrado espacial adaptativo, y funciona detectando y removiendo los componentes de alta frecuencia y alta amplitud en relación a una referencia de datos libre de artefactos, de esta manera se pretende recuperar la actividad de fondo del EEG que se encuentra en el subespacio abarcado por los componentes del artefacto (para más detalles revisar la referencia [44]). Un ejemplo de la aplicación de ASR se muestra en la figura 3.1.

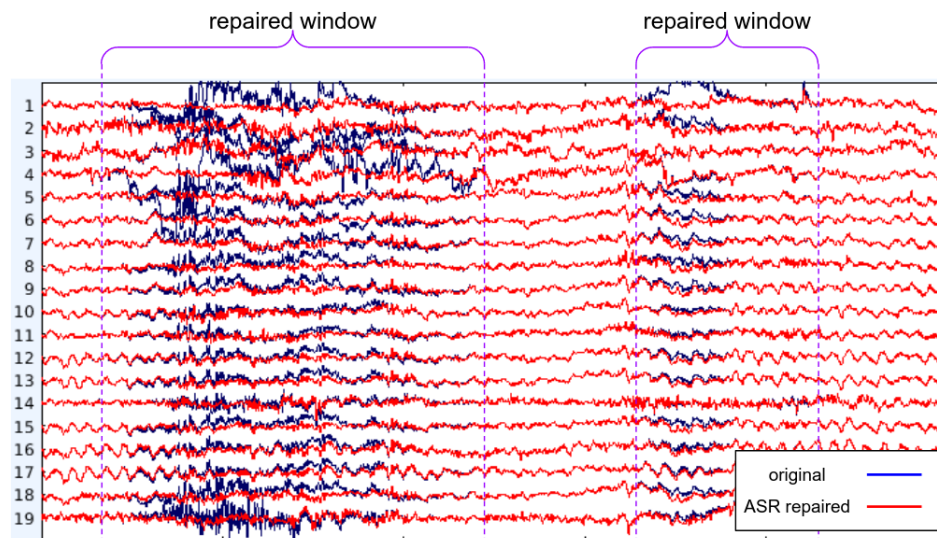


Figura 3.1: 5 segundos de una señal EEG luego de aplicar la limpieza de datos ASR (línea roja) superpuesta en la señal EEG original (línea azul).

Para que `clean_rawdata()` detecte los componentes no deseados de la señal, se deben fijar ciertos parámetros específicos y parámetros generales. En lo que respecta a esta tesis, se utilizan los parámetros generales por defecto, y los parámetros específicos se fijan con los valores a continuación:

- Remover canales que sean planos por más de 5 segundos.
- **No** aplicar filtrado.
- Remover canales que presenten menos de 0.8 de correlación con sus canales vecinos.
- Remover canales que presenten ruido de alta frecuencia con una desviación estándar mayor a 4.
- Reparar con ASR las ventanas que presenten ráfagas (comúnmente llamadas *bursts*) con una desviación estándar mayor a 20.
- Remover ventanas que presenten menos de 0.3 de sus canales aceptables (la definición de “aceptables” viene dada por los parámetros generales).

Cabe destacar que para esta tesis se utilizó una versión antigua (1.10) de `clean_rawdata()`. Esto debido a que el método posee una variante aleatoria y las versiones actuales no garantizan la replicabilidad de los resultados, a diferencia de la versión 1.10 que posee un parámetro ajustable para aquello.

6. **Interpolación de canales:** En caso de que la etapa anterior haya eliminado algún canal, se prosigue realizando una interpolación para recuperar aquellos canales faltantes. Para esto se utiliza un método de interpolación basado en el uso de superficies esféricas *spline* [45] de los canales, pues se requiere saber la ubicación de cada uno para realizar las ponderaciones correspondientes. De esta manera se computan los valores de los canales previamente removidos, si es que los hubo.

Algunos creen que no es bueno regenerar de esta manera los canales “malos” y que es mejor solo conservar los canales restantes, pero es un costo que en esta tesis se dispone

a pagar. La razón por las cuales se considera beneficioso realizar la interpolación son las siguientes:

- Se minimiza un potencial sesgo en la próxima etapa donde se realiza un *average referencing*. Por ejemplo, si en algún caso se hubieran rechazado solo canales del hemisferio derecho, luego la cantidad de canales en cada hemisferio dejaría de ser la misma, por lo que el *average* estaría sesgado hacia el hemisferio izquierdo.
- Como el objetivo es entrenar una red neuronal con los registros EEG, es conveniente que todos los registros presenten un formato lo más similarmente posible entre sí. Pues podría resultar confuso para la red recibir en algunas entradas, por ejemplo, dos canales menos o 2 canales fijados en cero, en comparación a otras entradas con más canales disponibles.

7. **Re-referenciación:** Los EEG requieren de una referencia con respecto a la cual computar las señales registradas, la cual suele variar según el conjunto de datos usado. Para el caso de esta tesis, como se mencionó en la anterior descripción de la base de datos, las señales se registraron siendo referenciadas al lóbulo derecho de cada paciente.

Una vez ya registradas las señales EEG, existen diferentes maneras válidas de re-referenciar estas. Particularmente en esta tesis se elige aplicar un re-referenciado *average*, el cual utiliza el promedio de todos los canales como señal de referencia; es decir, a cada señal correspondiente a un canal se le resta el promedio de todos los canales. Lo anterior resulta en un procedimiento simple que no requiere de información adicional a los datos registrados, además es independiente de la ubicación de referencia previa del registro.

Aplicar este re-referenciado resulta bastante útil para suprimir artefactos que aun permanecen en el registro, puesto que la presencia de estos suele manifestarse en varios canales a la vez, por lo cual al promediar se remueven parcial o completamente del resto de la señal.

8. **Submuestreo de la señal:** Como se mencionó en la descripción de la base de datos, las señales a utilizar fueron registradas con una frecuencia de muestreo de 1000 Hz. En este caso se realiza un *downsampling* a 500 Hz. Esto resulta útil para comprimir el tamaño de los datos y, por otro lado, sigue estando acorde al teorema del muestreo de Nyquist ($f_{sampling} \geq f_{max}$), pues los datos que se están procesando presentan una frecuencia máxima de 100 Hz debido a la previa aplicación de un filtro pasa-bajo.

Dado que en el enfoque con aprendizaje profundo se aprovechará el uso de transferencia de aprendizaje, cabe destacar que será necesario volver a aplicar *downsampling* en las señales antes de ingresarlas al modelo, puesto que estas deben estar acordes al formato de las señales con las cuales se realizó el preentrenamiento (para más detalles ver sección 3.4).

Para ilustrar la *pipeline* descrita, en la figura 3.2 se muestra un extracto de señal EEG antes de ingresar a la red neuronal del paso 4, y sobre aquella, se dispone la misma señal después de la re-referenciación del penúltimo paso. De esta manera se observa como la *pipeline* es capaz de remover componentes de alta frecuencia o alta amplitud conservando las bandas más valiosas.

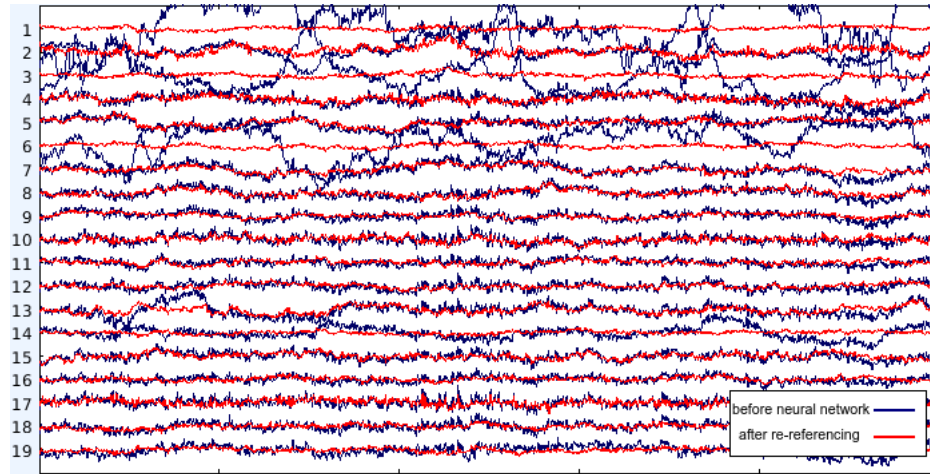


Figura 3.2: 5 segundos de una señal EEG justo antes de ingresar a la red neuronal del paso 4 (línea azul), y justo después de salir del re-referenciado del paso 7 (línea roja).

3.2.2. Etiquetado de las señales

Aunque el objetivo inicial de esta tesis era obtener una estimación del nivel de compensación/descompensación clínica de cada paciente, por razones de simplicidad y robustez, se optó por modificar este objetivo a la obtención de una estimación del nivel de síntomas positivos presentados por el paciente en un instante dado. Recordar que los síntomas de la esquizofrenia se suelen dividir en dos subgrupos: síntomas positivos y síntomas negativos, para más detalles ver sección 2.1.

Cada registro de EEG se etiquetó como “síntomas positivos elevados” (denominado clase “1”) o “síntomas positivos regulares/bajos” (denominado clase “0”), referente al estado del paciente al momento de registrar las señales. Para determinar la etiqueta de cada registro se analizaron los metadatos del estado clínico, sobre los cuales se aplicó un criterio específico, el cual se detalla a continuación.

Dentro de la información contenida en la metadata, se puso atención en los puntajes **PANSS** (*Positive and Negative Syndrome Scale*) y **SAPS** (*Scale for the Assessment of Positive Symptoms*). Ambas escalas evalúan al paciente por medio de una entrevista de aproximadamente 40 a 50 minutos, donde el entrevistador debe estar capacitado para un mínimo nivel estándar de confiabilidad [46]. A continuación se explica en qué consisten estas escalas,

- **PANSS**: Mide la severidad de los síntomas, en general, de los pacientes con esquizofrenia; es ampliamente usada en el estudio de la terapia de antipsicóticos. Se le conoce como el “estándar de oro” que deben seguir todas las evaluaciones de los trastornos psicóticos del comportamiento [47].

El paciente es calificado de 1 (menos presente) a 7 (más presente) en 30 ítems (síntomas) diferentes, basándose tanto en la entrevista como en reportes de miembros de la familia o trabajadores del hospital de atención primaria. En esta escala los síntomas se subdividen en 3 grupos: positivos, negativos y de psicopatología general. El puntaje PANSS

total está dado por

$$PANSS = \sum_{i=1}^7 P_i + \sum_{i=1}^7 N_i + \sum_{i=1}^{16} G_i, \quad (3.1)$$

donde P_i representa los síntomas positivos, N_i los negativos y G_i los de psicopatología general.

- **SAPS**: Mide la severidad solo de los síntomas positivos. La escala se divide en 4 dominios (alucinaciones, delirios, comportamiento bizarro y desorden del pensamiento formal positivo), y dentro de cada dominio diferentes ítems (síntomas) son calificados de 0 (ausente) a 5 (severo) [48]. Los puntajes no se suman, pues la escala suele ser usada para analizar cada uno de manera individual.

Como se puede apreciar, ambas escalas evalúan a su manera los síntomas positivos presentes en la esquizofrenia, y como es de esperar, se ha mostrado que ambas poseen una correlación importante entre sí [49, 50]. Por otro lado, las mediciones de ambas escalas son dependientes de la percepción o interpretación del entrevistador, por lo que son métricas subjetivas, lo cual implica que son menos fiables pues un mismo estado no siempre dará como resultado un mismo puntaje. Es por lo anterior, que en el proceso de etiquetado de las señales, se decidió considerar tanto el puntaje PANSS como el puntaje SAPS de manera conjunta, en vez de cada uno por sí solo. Así se pretende agregar un mayor nivel de robustez al proceso de decisión de si en un momento específico el sujeto en cuestión se encuentra con niveles de síntomas positivos altos o niveles de síntomas positivos regulares/bajos.

Para cada registro EEG se tienen dos puntajes a considerar: el PANSS y el SAPS. Para definir el criterio utilizado para etiquetar los registros se decidió unificar ambos puntajes en un solo valor. Para ello, se aprovecha el análisis de regresión realizado en [51], de donde se deriva la siguiente ecuación de conversión

$$\widehat{SAPS}_{global} = -3.222 + (0.567 * PANSS_{positive}), \quad (3.2)$$

donde $SAPS_{global}$ corresponde a la suma de 4 ítems del SAPS (7, 20, 25 y 34, específicamente) que corresponden al puntaje de calificación global en cada uno de los 4 dominios, y \widehat{SAPS}_{global} representa la estimación de dicha suma; mientras que $PANSS_{positive}$ corresponde a la suma de los 7 ítems de síntomas positivos del PANSS. En otras palabras, a partir de los síntomas positivos proporcionados por el PANSS, la ec. (3.2) entrega una estimación de la equivalencia en síntomas positivos que debiera proporcionar la suma de los ítems globales del SAPS.

Previo a utilizar la ec. (3.2) dentro del criterio de etiquetado, como *sanity-check* se verifica visualmente que los puntajes, $SAPS_{global}$ y $PANSS_{positive}$, reportados por los metadatos si posean una relación semejante a la representada por la ecuación. Para esto se grafican aquellos datos en la figura 3.3, y sobre aquellos se grafica la recta generada por (3.2); de donde se confirma una cierta relación.

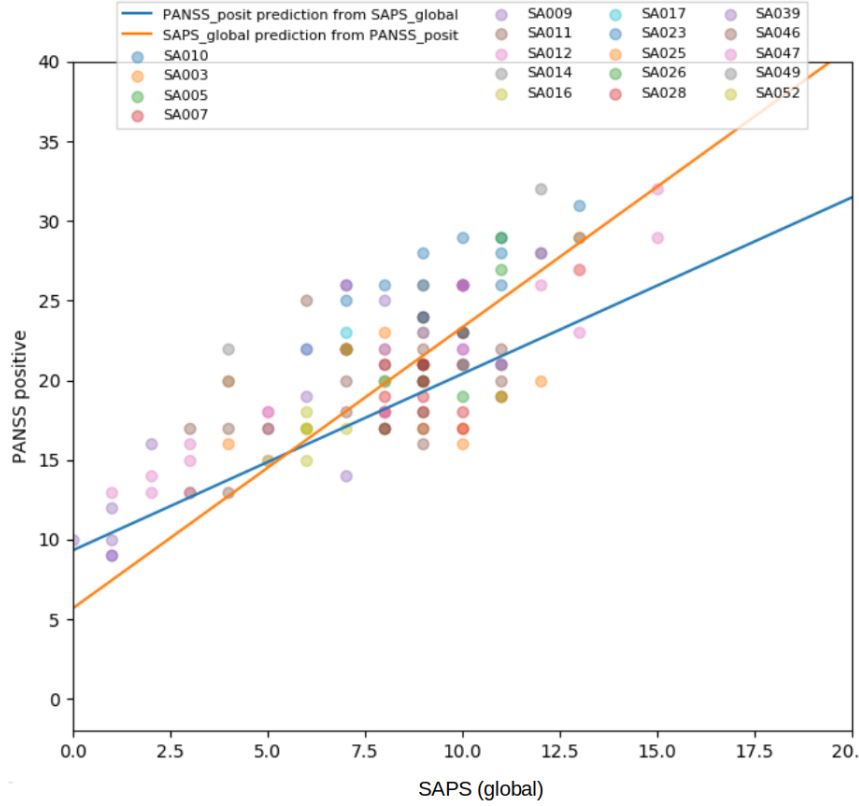


Figura 3.3: Puntajes $SAPS_{global}$ y $PANSS_{positive}$ reportados por los metadatos gráficamente en puntos. Mientras que se sobreponen las rectas generadas por las ecuaciones $\widehat{SAPS}_{global} = -3.222 + (0.567 * PANSS_{positive})$ (recta anaranjada) y $PANSS_{positive} = 9.3264 + (1.1072 * SAPS_{global})$ (recta azul), introducidas por análisis regresivo en [51].

Teniendo en cuenta lo anterior, se decidió utilizar el subpuntaje $SAPS_{global}$ como criterio para realizar el etiquetado de las señales, pero además, se aprovecha la estimación \widehat{SAPS}_{global} para agregar mayor robustez al criterio. Entonces, para cada señal EEG se considera un valor $SAPS_{global}$, obtenido directamente del puntaje SAPS reportado en los metadatos asociados, y un valor \widehat{SAPS}_{global} , el cual se obtiene por medio de la ec. (3.2) usando el puntaje PANSS reportado en los metadatos asociados.

Finalmente, se obtiene para cada registro EEG, un solo valor definitivo que se denomina $SAPS_{def}$, el cual está dado por

$$SAPS_{def} = \frac{SAPS_{global} + \widehat{SAPS}_{global}}{2}. \quad (3.3)$$

Con el valor final $SAPS_{def}$ se pueden etiquetar los registros de dos maneras:

- Utilizar un **target continuo**, $label_cont_{rec_{ij}}$, el cual será utilizado en el enfoque con extracción manual de características, y por ende, usar directamente $SAPS_{def}$ como etiqueta, es decir

$$label_cont_{rec_{ij}} = SAPS_{def}. \quad (3.4)$$

- Utilizar un **target por categoría**, $label_categ_{rec_{ij}}$, en este caso binaria, el cual será

utilizado en el enfoque con aprendizaje profundo, y por ende, se debe fijar un umbral para aquello.

Para la segunda alternativa se debe calcular un umbral fijo, el cual debido a la naturaleza heterogénea de la EQZ (ver sección 1.1), será diferente para cada sujeto. El umbral $\mu_{i,label}$, para un sujeto i , está dado por

$$\mu_{label} = \text{median}(\{SAPS_{def,j}\}_{j=0}^M), \quad (3.5)$$

donde M es la cantidad de registros disponibles para el sujeto i . En resumen, para un sujeto i , se prosigue a etiquetar binariamente cada registro rec_{ij} según el umbral asignado, es decir

$$label_categ_{rec_{ij}} = \begin{cases} 1 & \text{if } SAPS_{def,j} > \mu_{i,label} \\ 0 & \text{if } SAPS_{def,j} \leq \mu_{i,label} . \end{cases} \quad (3.6)$$

Gráficas ilustrativas acerca del proceso de etiquetado de las señales recién descrito, son proporcionadas en la sección A del apéndice.

Finalmente, cabe recordar que para el ajuste de los modelos que más adelante se describen, se utilizarán **segmentos** de 40s de los registros. Con respecto al etiquetado de estos segmentos, se decide que cada segmento tendrá la etiqueta que le corresponde al registro del cual forma parte.

3.2.3. Base de datos auxiliar

Además de la base de datos inédita descrita anteriormente, en esta tesis se utiliza, en ciertas etapas, una base auxiliar; esto con el objetivo de chequear el funcionamiento de algunos métodos o incluso aprovechar de probar transferencia de aprendizaje. Este conjunto de datos tiene la característica de ya haber sido analizado en trabajos anteriores [52–55]. Por otra parte, a diferencia de la base de datos descrita anteriormente, la cual busca discriminar entre sujetos con sintomatología positiva alterada y sintomatología positiva regular, esta base de datos auxiliar se registró con el objetivo de discriminar entre sujetos con EQZ y sujetos *healthy-control* (HC). Por ende, la etiqueta asociada a cada registro EEG de esta base de datos auxiliar, está dada simplemente conociendo a qué tipo de sujeto se le tomó el registro.

Esta base de datos alternativa corresponde a señales EEG de 14 pacientes con EQZ (7 hombres: 27.9 ± 3.3 años, 7 mujeres: 28.3 ± 4.1 años), la cual fue recolectada en el *Institute of Psychiatry and Neurology in Warsaw* de Polonia [56]. Pacientes con algún otro tipo de dolencias neurológicas (e.g. Alzheimer, epilepsia, etapa de EQZ muy temprana), embarazo o algún otra condición médica general fueron excluidos. Además, se recolectaron señales EEG de otros 14 sujetos sanos, los que se encuentran dentro del mismo grupo etario y proporción de género.

Los registros EEG en ambos casos fueron recolectados en estado reposo con ojos cerrados. Estos fueron recolectados a una frecuencia de muestreo $f_s = 250\text{Hz}$ en 19 canales del cuero cabelludo ($Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2$), posicionados de acuerdo al *International 10-20 system*. El electrodo de referencia fue posicionado en FCz . Cabe mencionar, que estas señales no han sido preprocesadas, por lo que al igual que los datos antes descritos, deben ser limpiadas. Para efectos de esta tesis, esta base de datos se denomina **EQZvsHC**.

Estos registros también serán procesados por segmentos de 40 s, y de igual manera la etiqueta de cada segmento corresponderá al registro del cual forma parte.

3.3. Enfoque con extracción manual de características

El primer método propuesto, como ya se adelantó, consiste en extraer características de las señales EEG, posterior a su limpieza, y en lo posible analizar estas por medio de aprendizaje de máquinas. La particularidad, en este caso, es que las características extraídas son previamente elegidas por seres humanos, y por ende, también se debe explicitar cómo extraerlas.

En el aprendizaje de máquinas la elección de las características a extraer es una tarea crítica en el proceso de modelado, puesto que las características seleccionadas poseen un gran impacto en la precisión del modelo resultante. En lo que respecta a la tarea de esta tesis, como se menciona en el capítulo 1, no existe evidencia fuerte que indique qué aspectos del EEG pueden resultar relevantes para discriminar entre los diferentes estados clínicos por los que transiciona cada paciente; a excepción de lo reportado en [8, 9], donde se señala que el aumento de ruido del registro EEG estaría correlacionado con los períodos de exacerbación sintomática del paciente. Por todo lo anterior no resulta trivial escoger las características a extraer.

3.3.1. Características a extraer

Mediante conversaciones con investigadores del área y revisión bibliográfica se seleccionaron las siguientes características, que se cree, podrían entregar información acerca del nivel de estabilidad clínica de los pacientes con EQZ. Las características a extraer serán todas sobre vectores (señales) de 1 dimensión, y se explican a continuación. Dado una señal $X \in \mathcal{R}^{T \times 1}$, con un T arbitrario, se define:

1. **Promedio:** Suma del conjunto de valores dividida por el largo total de la señal:

$$Promedio(X) = \frac{1}{T} \sum_{i=1}^T x_i = \frac{x_1 + x_2 + \dots + x_T}{T}. \quad (3.7)$$

2. **Mediana:** Al ordenar todos los valores de la señal en orden creciente, la mediana representa el valor de la variable de posición central. Si el largo de la serie es par, luego la mediana sería el promedio entre los dos valores de posición central.

3. **Desviación estándar (STD):** Cuantifica la variación del conjunto de valores de la señal,

$$STD(X) = \sqrt{\frac{\sum_i (x_i - Promedio(X))^2}{T}}. \quad (3.8)$$

4. **Kurtosis:** Cuantifica la forma de la distribución de frecuencias de los valores de la señal. En este caso particular se utiliza el “exceso de *kurtosis*” [57], el cual mide qué tanto difieren las “colas” de la distribución con respecto a las “colas” de una distribución normal (ver figura 3.4).

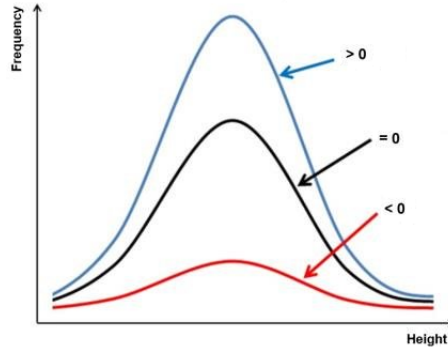


Figura 3.4: Ilustración del exceso de *kurtosis* en una distribución.

El “exceso de *kurtosis*” se obtiene por la siguiente ecuación

$$Kurt(X) = \frac{\sum_i^T \frac{(x_i - Promedio(X))^4}{T}}{STD(X)^4} - 3. \quad (3.9)$$

5. **Skewness**: Cuantifica la asimetría de la distribución de frecuencia de los valores de la señal (ver figura 3.5). Para una distribución normal, el *skewness* debería ser cercano a cero.

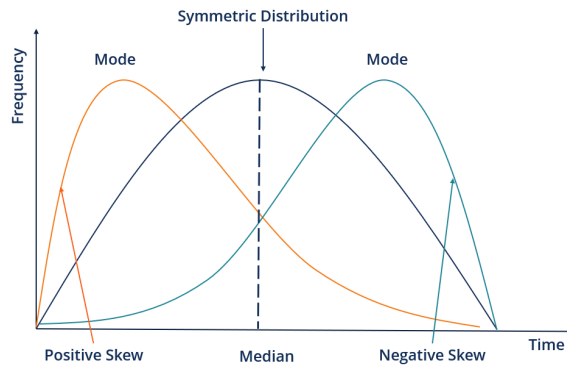


Figura 3.5: Ilustración del *skewness* de una distribución.

El *skewness* se obtiene mediante la siguiente ecuación [57]

$$Skew(X) = \frac{\sum_i^T \frac{(x_i - Promedio(X))^3}{T}}{\left(\sum_i^T \frac{(x_i - Promedio(X))^2}{T}\right)^{3/2}}. \quad (3.10)$$

6. **Rango intercuartílico (IQR)**: Es una medida de dispersión, y corresponde a la diferencia entre el tercer y primer cuartil de una distribución. Donde el primer cuartil (Q_1) corresponde a la mediana de la primera mitad de los valores de la señal ordenados, y el tercer cuartil (Q_3) corresponde a la mediana de la segunda mitad de los valores ordenados. Luego

$$IQR(X) = Q_3(X) - Q_1(X). \quad (3.11)$$

7. **Potencia promedio**: Estrictamente, la potencia de una señal es la suma de los cua-

drados absolutos de sus valores a lo largo del tiempo, dividida por el largo de la señal. Pero en este caso particular, no se considerará el cuadrado, sino que se tomará el valor absoluto de la señal, puesto que se usará para comparar valores y sigue siendo útil para aquello. Luego se define como

$$PotProm(X) = \frac{1}{T} \sum_{i=1}^T |x_i|. \quad (3.12)$$

8. **Mediana de la desviación absoluta (MAD):** Mide la variabilidad del conjunto de valores de la señal, al igual que la STD. La diferencia es que el MAD es más robusto, pues se ve menos afectado por la presencia de valores extremadamente altos o extremadamente bajos. El MAD suele utilizarse para conjuntos de valores que no están normalmente distribuidos. Su valor se define como la mediana de las desviaciones absolutas de la mediana de la señal, es decir

$$MAD(X) = Median(|x_i - Median(X)|). \quad (3.13)$$

9. **Complejidad de Lempel-Ziv (LZC):** Medida no lineal usada para cuantificar el nivel de aleatoriedad o ruido de una señal; para más detalles al respecto ver sección 2.3. Para su cálculo, se transforma previamente la señal de EEG a información binaria. Los criterios utilizados para binarizar la señal son variados, donde generalmente se utiliza un umbral fijo que viene dado por la mediana de la señal. En este caso particular se decide utilizar diferentes criterios, y de esta manera, se obtienen diferentes submétricas basadas en la LZC, donde el algoritmo para computar la complejidad será el mismo, solo variará la elección del umbral usado en el proceso de binarización previo. Para cada caso se fija el umbral μ , y dada una señal S , la binarización está dada por la ec. (2.1). Dado que varias submétricas dependen de la LZC en este análisis, es que se ejecutan algunos *sanity checks* del algoritmo de LZC implementado para este trabajo, para asegurar que la métrica efectivamente esté cuantificando de manera correcta la aleatoriedad de la señal; para ver estos resultados se puede revisar la sección B del apéndice.

Se establecen las siguientes características basadas en LZC, donde el origen del umbral fijado para binarizar en cada caso se describe a continuación:

- a) **LZC-median:** Se utiliza la mediana de la señal como umbral fijo para binarizar.
- b) **LZC-Hilbert:** Se aplica la transformada de Hilbert $H(\cdot)$ a la señal y de esta manera se obtiene la llamada señal analítica $H(S)$, desde donde al tomar su magnitud, se puede extraer una aproximación de la envolvente de la señal. Luego, se toma la mediana de la envolvente extraída, $|H(S)|$, y aquello se utiliza como umbral fijo para binarizar.
- c) **LZC-Hilbert-pow.:** Se aplica la transformada de Hilbert $H(\cdot)$ a la señal y de esta manera se obtiene la llamada señal analítica $H(S)$, luego se obtiene el cuadrado de esta, desde donde se toma la magnitud. Luego, se toma la mediana de este resultado, es decir, de $|H(S)^2|$, y aquello se utiliza como umbral fijo para binarizar.
- d) **LZC-slope:** A cada valor de la señal $S(x_i)$ se le resta el siguiente valor $S(x_i + 1)$, obteniéndose así el valor de la pendiente de la señal en cada punto. Luego, se binariza

cada punto de la señal considerando si la pendiente es positiva o negativa en aquél punto. Visto de otra manera, se fija un umbral 0 y se revisa en cada punto el valor de la pendiente.

- e) **LZC-slope-Hilbert**: Se obtiene la misma aproximación de la envolvente de la señal obtenida en (b). Luego a cada valor $|H(S(x_i))|$ se le resta el siguiente valor $|H(S(x_i + 1))|$, obteniéndose el valor de la pendiente de $|H(S)|$ en cada punto. Luego, se binariza cada punto de la señal considerando si la pendiente es positiva o negativa en aquél punto. Visto de otra manera, se fija un umbral 0 y se revisa en cada punto el valor de la pendiente.
- f) **LZC-slope-Hilbert-pow.**: Se obtiene la misma expresión obtenida en (c). Luego a cada valor $|H(S(x_i))^2|$ se le resta el siguiente valor $|H(S(x_i + 1))^2|$, obteniéndose el valor de la pendiente de $|H(S)^2|$ en cada punto. Luego, se binariza cada punto de la señal considerando si la pendiente es positiva o negativa en aquél punto. Visto de otra manera, se fija un umbral 0 y se revisa en cada punto el valor de la pendiente.
- g) **LZC-mean**: Se utiliza el promedio de la señal como umbral fijo para binarizar.

En la figura 3.6 se ilustran los criterios utilizados para fijar el umbral en cada caso. El caso de *LZC-mean* se omite, pues al ser casi similar a *LZC-median*, resulta redundante ilustrarlo.

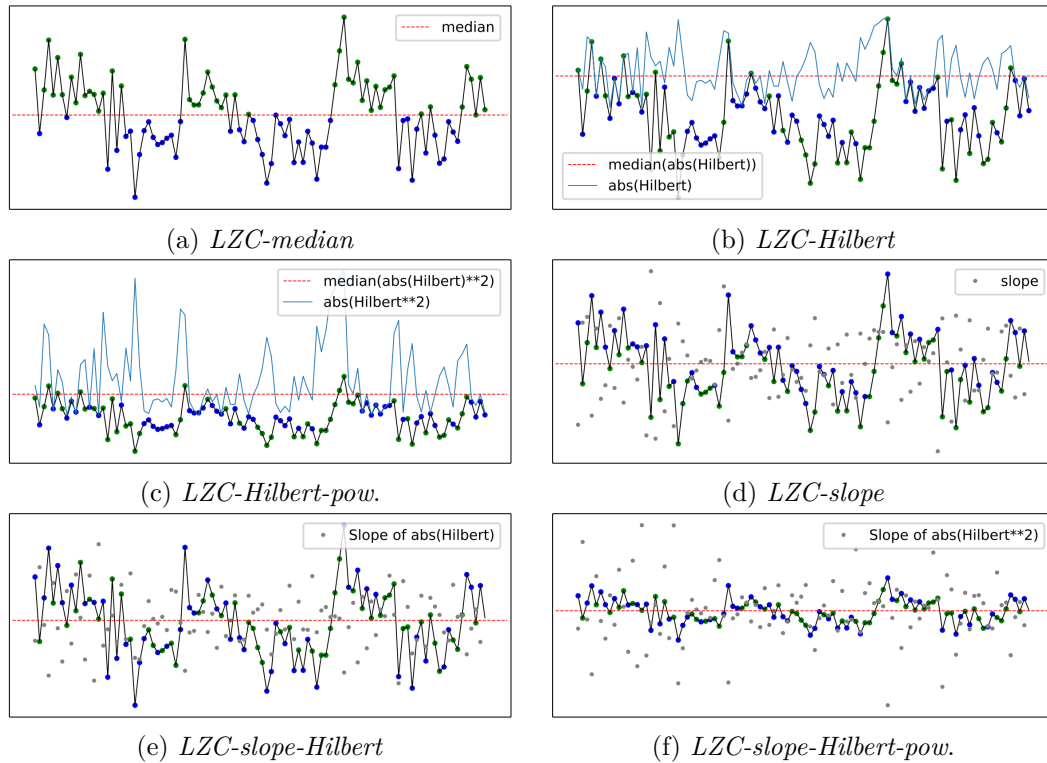


Figura 3.6: Ilustración de umbral usado en cada método de binarización previo al cómputo de la LZC. La línea punteada roja indica el umbral considerado para que los valores en azul sean codificados como ‘1’s y los valores en verde sean codificados como ‘0’s. Notar que en cada gráfico se plotea la misma señal pero a diferente escala, con el fin de observar mejor los criterios utilizados para establecer el umbral.

3.3.2. Extracción y análisis de características

Con respecto a cómo serán manipulados los registros EEG, debido a que la cantidad de registros disponibles es reducida, estos serán segmentados en ventanas de 40 s con un traslape de 30 s. Por otro lado, para disminuir costos computacionales y por simplicidad, se decide en este caso solo considerar 1 canal de los registros, el canal Cz, el cual se elige debido a su centralidad y menor cantidad de ruido. Por lo tanto, la extracción de características se realizará sobre cada ventana y en un solo canal, lo que se denominará **un segmento**. Como la frecuencia de muestreo de los registros es $f_s = 500Hz$, la extracción de características se realizará sobre cada segmento $x_{seg} \in \mathcal{R}^{(500 \cdot 40) \times 1} = \mathcal{R}^{20000 \times 1}$.

Además de calcular las características para cada registro EEG, se calculan las mismas características para los registros filtrados en bandas, de tal manera de aumentar aun más el número de características disponible. Es decir, para cada registro EEG se calculan las 15 características descritas en la sección 3.3.1. Luego, el registro es filtrado en las bandas alfa, beta, gamma, delta y theta; donde a cada banda se le calculan las mismas 15 características. En resumen, para cada registro EEG se cuenta con 90 características.

Por otro lado, un número tan elevado de características puede resultar contraproducente para un número tan bajo de segmentos o muestras de EEG disponibles. Por lo anterior, por medio de un método basado en Información Mutua implementado por Juan Urrutia [58], se procede a reducir el número de características según el caso. Este método de selección de características analiza las informaciones mutuas individuales y colectivas, para encontrar el subconjunto óptimo de características.

El método de Juan Urrutia analizó todas las características extraídas, de cada segmento, considerando como *target* las etiquetas de estos segmentos. El algoritmo se ejecutó 2 veces, una considerando las etiquetas categóricas (ec. (3.6)), y otra considerando las etiquetas continuas (ec. (3.4)) normalizadas **por sujeto** entre -1 y 1. Cabe mencionar que esta normalización por sujeto se sugiere debido a que el rango de valores que toma $SAPS_{def}$ es muy variable dependiendo el sujeto; y esto es debido a que cada sujeto posee un “estado base” de compensación diferente.

Una vez ejecutado el método de selección de características, este retorna conjuntos de características sugeridos, los cuales debiesen tener la cualidad de poseer una alta sinergia y una baja redundancia, dentro de lo posible, con respecto a los *targets* definidos. Varios conjuntos de características se evaluaron realizando gráficos de dispersión de 2D y 3D. Si el conjunto resultante tiene más de 3 características, se usa el algoritmo *Uniform Manifold Approximation and Projection for Dimension Reduction* (UMAP)⁶, el cual es capaz de reducir el espacio a 2D y así facilitar su visualización [59]. Los parámetros de UMAP fueron ajustados por *trial-and-error*, donde finalmente se elige utilizar un tamaño de vecindad de 5 y utilizar la métrica de distancia de Mahalanobis.

La mayoría de los conjuntos de características sugeridos poseen más de 3 características, por lo que analizar la interacción de estas solo visualmente con gráficos 2D resulta muy limitado, aun incluso utilizando UMAP. Es por esta razón que, además, se analizan algunos

⁶ Algoritmo para reducir dimensionalidad por medio del uso de estructuras topológicas difusas.

conjuntos de características por medio de modelos *random forest* (RF); esto con el objetivo de explorar las interacciones, entre características, que no sean explícitamente visibles en gráficas de 2 dimensiones.

3.4. Enfoque con aprendizaje profundo

Se propone un método para analizar las señales EEG en crudo, pero siempre con la previa aplicación de la limpieza mencionada en la sección 3.2.1. En este caso, en crudo se refiere a que las señales mismas son las que se analizan directamente, sin la necesidad de extraer manualmente alguna característica.

Como se mencionó en la sección 2.4.4, el modelo profundo debiera ser capaz de encontrar, a través del entrenamiento, una manera efectiva de representar los datos (i.e., extraer características) para lograr resolver la tarea solicitada, que en este caso corresponde a la discriminación entre síntomas positivos elevados o síntomas positivos regulares/bajos.

El entrenamiento de un modelo de aprendizaje profundo implica ajustar un elevado número de parámetros, por lo que se suele requerir de una cantidad considerable de datos para aquello. Dado que en este caso la cantidad de sujetos y registros EEG es limitada, es que se decide abordar el enfoque con transferencia de aprendizaje. Dicho de otra manera, se pretende, previo al entrenamiento, usar parámetros iniciales en el modelo que se hayan obtenido entrenando con una gran cantidad de datos similares. Para lograr lo anterior es necesario reutilizar una arquitectura, o parte de ella, que ya haya sido preentrenada con un conjunto de datos similares o relacionados a los datos que se utilizarán luego para afinar el modelo.

3.4.1. Arquitectura

Para la transferencia de aprendizaje, se decide utilizar una arquitectura que ya ha sido entrenada con una gran cantidad de registros EEG. Para ello, se utiliza lo propuesto en el paper “*BENDR: Using Transformers and a Contrastive Self-Supervised Learning Task to Learn From Massive Amounts of EEG Data*” [60]. En este trabajo se propone adaptar una arquitectura diseñada originalmente para modelamiento del lenguaje, y entrenarla con una gran cantidad de datos con el objetivo de desarrollar un modelamiento de EEGs en la misma línea. Los autores adaptan específicamente un enfoque de entrenamiento autosupervisado, con el cual el modelo aprende a representar de manera comprimida las señales en crudo en un espacio latente intermedio. Finalmente, muestran que tanto la representación de los datos aprendida por el modelo como la arquitectura completa puede ser ajustada (*fine-tuned*) a una variedad de tareas de clasificación de EEG, independiente del *hardware* usado para registrar los datos, o del uso de diferentes sujetos para diferentes tareas. Los pesos obtenidos de este preentrenamiento se encuentran públicamente disponibles [61], y son los que se aprovechan en esta tesis.

Para que los pesos de preentrenamiento recién mencionados sean útiles, se establecen algunos parámetros de referencia en el paper, esto con el objetivo de que independiente del sujeto o equipo de *hardware* de donde sean extraídas las señales EEG, éstas puedan ser transversalmente representadas con un mismo modelo o, que sean fácilmente representables

aplicando un breve *fine-tuning*, es decir, aplicando transferencia de aprendizaje. En otras palabras, se debe estandarizar cómo y en qué condiciones debe estar un registro EEG antes de ser ingresado a la red, ya sea para probar su desempeño o para aplicar *fine-tuning*. Para esto se establece un breve preprocesamiento que debe ser aplicarse a la señales, de esta manera se pretende estandarizar lo más posible las condiciones en que ingresan señales EEG de diferentes fuentes o condiciones. Por ahora se mencionará, a grandes rasgos, que las señales que van a ingresar deben presentar una frecuencia de muestreo de $f_s = 256Hz$, 19 canales de EEG, y además, deben ser normalizadas entre $[-1,1]$. Por esta razón se les adicionará un canal extra a los registros para describir la escala relativa de la señal con respecto al resto del conjunto de datos, es decir, la señal resultante a ingresar poseerá 20 canales. Todo este preprocesamiento recién mencionado se describe en detalle en la sección 3.4.1.1.

Continuando con la arquitectura propuesta en [60], esta consiste en dos etapas: codificador y contextualizador. Se prueban 2 alternativas de contextualizador, uno lineal más sencillo, y otro más complejo que está compuesto principalmente por un *transformer encoder* [62]. En esta tesis se trabaja, por simplicidad y como primer enfoque, solo con el contextualizador lineal, modificando parámetros y algunas capas. Entonces, la arquitectura a usar en esta tesis se compone de las siguientes etapas:

- **Codificador:** A grandes rasgos, esta etapa recibe las señales EEG en crudo y las procesa por 6 bloques convolucionales de 1-D consecutivas, así reduce considerablemente la dimensión (*downsampling*) a una nueva secuencia de vectores.

Más específicamente, la señales en crudo son reducidas por medio del tamaño de *stride* de las capas, pues se requiere de menos memoria que al reducir con *pooling*. Cada bloque convolucional está compuesto por:

- Convolución 1-D
- *GroupNorm* [63]
- GELU [40]

Como ya se mencionó, el codificador está compuesto por 6 bloques consecutivos, cada uno con kernel de tamaño 2, excepto el primer bloque que posee kernel de tamaño 3. El tamaño de los *strides* de cada bloque corresponderá al mismo tamaño de kernel por bloque (es decir: 3, 2, 2, 2, 2, 2). De esta manera, se tiene que la “frecuencia de muestreo efectiva” será 96 veces más pequeña que la frecuencia de muestreo original que ingresa al codificador. Por otro lado, cada bloque consiste en 512 filtros, por lo que el vector resultante tiene un largo de 512.

Considerando que las entradas al modelo son segmentos de largo 40 *s* y 20 canales, se tiene que, al considerar la frecuencia estándar de $f_s = 256Hz$, la entrada al modelo es de dimensión $(bs, 20, 10240)$. Luego, la dimensión de salida del codificador será de $(bs, 512, \frac{10240}{96}) = (bs, 512, 107)$. A esta salida se le llama **señal en estado latente**, z .

- **Contextualizador lineal:** A grandes rasgos, esta etapa recibe la señal en estado latente y se encarga de mapearla a una nueva secuencia que encarna la tarea de clasificación. El contextualizador se divide en 2 subetapas.

La primera subetapa se denomina **bloque de *encoding augment***, el que inicialmente, cuenta con una capa que actúa como un *embedding* de posición relativa. Esta consiste en una capa de convoluciones agrupadas, con kernel de tamaño 25 y, 16 grupos. Esta recibe la señal en estado latente como entrada, generando su propia salida, la cual pasa por una GELU y luego es sumada a la misma señal en estado latente. Luego se aplica *layer normalization*, y una capa convolucional más para triplicar el número de canales y generar una salida de dimensión $(bs, 512 * 3, 107) = (bs, 1536, 107)$.

A la salida del bloque de *encoding augment*, se aplica un *average pooling* de 1D, que reduce las secuencias a largo 4, quedando la salida con dimensión $(bs, 1536, 4)$.

Finalmente, se ingresa al denominado **bloque de *extended classifier***, en donde se aplica *flatten* a la señal, quedando esta con dimensión $(bs, 1536 * 4) = (bs, 6144)$. Originalmente, en [60], la señal se ingresa a una capa *fully connected* seguida de una ReLU y *batch normalization*; pero para disminuir el riesgo de sobreajuste, por *trial-and-error*, se decidió intercambiar la *fully connected* por otra capa *average pooling*, la cual disminuye considerablemente la cantidad de parámetros de la red. De esta manera se tiene un vector de dimensión $(bs, 2048)$, el cual se ingresa a una última capa *fully connected* que funciona como el clasificador final, el cual tiene una salida de dimensión $(bs, 1)$, la que corresponde al *logit* o puntaje final de la señal ingresada al modelo.

En la figura 3.7 se ilustra espacialmente la arquitectura arriba descrita. Como se puede observar, se utiliza la función de activación GELU [40] en el codificador y dentro del bloque de *encoding augment*, en vez de la función ReLU tradicional. Esto debido a su curvatura creciente y su no-monotonidad que le permite aproximarse más fácilmente a funciones más complejas, además de algunas otras ventajas que se mencionaron en la sección 2.4.6.

Cabe mencionar, que se incluye una técnica de regularización la cual consiste en enmascarar secciones de la salida del codificador, es decir, de la señal en estado latente. La idea es que solo durante el entrenamiento, se enmascaren secciones contiguas del 5% de la señal en estado latente completa, con una probabilidad de 0.01. Adicionalmente, a lo largo de la señal en estado latente (i.e. a lo largo de cada vector de la secuencia), se realiza un procedimiento similar “apagando” *features* a 0, donde secciones contiguas del 10% de los canales son descartadas con una probabilidad de 0.005.

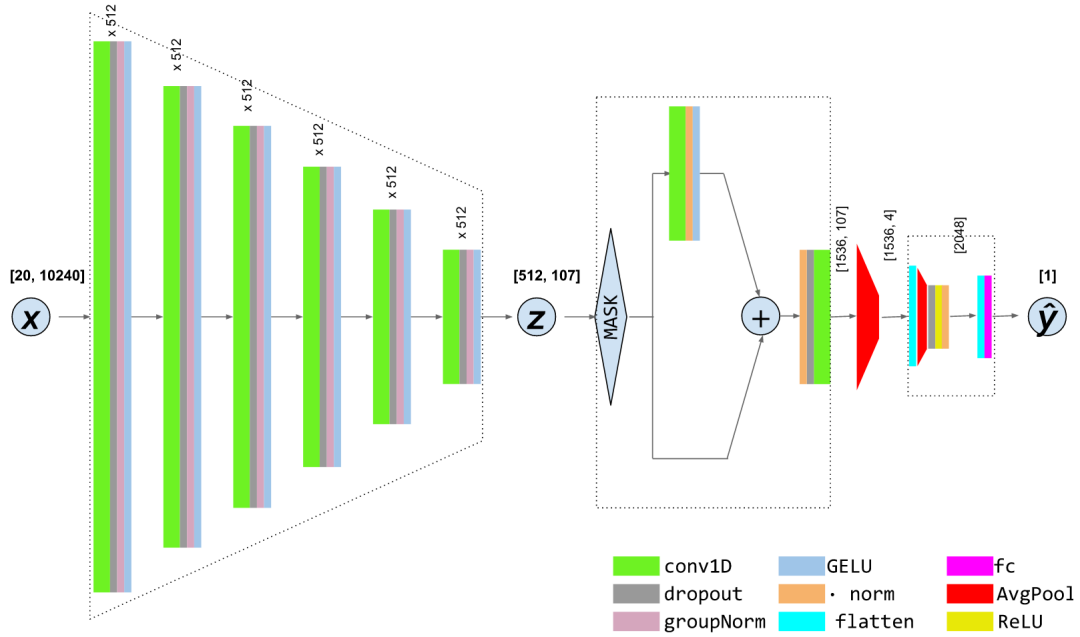


Figura 3.7: Arquitectura del modelo propuesto para aprendizaje profundo. De izquierda a derecha con líneas punteadas se indican los bloques: Codificador, *Encoding augment* y *Extended classifier*.

Otra de las técnicas de regularización incluidas, corresponde a la aplicación de dropout posterior a casi todas las capas. Dentro del codificador, por *trial-and-error* se obtiene la heurística de utilizar un dropout de 0.3, mientras que en la salida del *encoding augment* se usa dropout 0.1 y, finalmente, en el *extended classifier* se usa un dropout de 0.7.

3.4.1.1. Preprocesamiento de la señal

Tal como se mencionó en la descripción de la arquitectura, se aprovechará la disponibilidad de los pesos de preentrenamiento obtenidos en [60] y, con ello, se hará uso de *transfer learning*. Para que esto funcione, es necesario modificar levemente las señales EEG a usar para que estas estén con la misma configuración y orden con las que se preentrenó el modelo. De hecho, la enorme cantidad de registros EEG con los que se preentrenó en el paper, también fueron estandarizados con el mismo método. A continuación, se describe el preprocesamiento aplicado a cada registro EEG. Cabe destacar, que este preprocesamiento no debe confundirse con el proceso de limpieza explicado en la sección 3.2.1; son procesos diferentes, donde la limpieza se aplica primero, y luego, se aplica este preprocesamiento a la señal antes de ser ingresada a la red.

El preprocesamiento para estandarizar las señales, y que estén en sintonía con las señales con las que se preentrenó el modelo, consiste en las siguientes etapas:

1. **Transformación *MappingDeep1010***: Un orden consistente de los canales es necesario para permitir ingresar al modelo datos obtenidos por diferentes *hardwares*, y además, así permitir la transferencia de aprendizaje para diferentes aplicaciones con diferentes señales EEG. Un orden consistente se refiere a que el mismo canal de EEG estará siempre representado por el mismo índice del tensor de datos, e.g. el canal *FP1* siempre se encuentra en el índice 1. Para hacer esto automáticamente se utiliza el sistema *Mapping-Deep1010* implementado en la librería DN3 [64], el cual realiza un mapeo consistente de

77 canales EEG; en este caso el conjunto de datos cuenta con 19 canales por lo que solo mapea 19 canales, el resto queda en 0.

2. **Interpolación temporal:** Aplica *downsample/upsample* a la señal para que esta coincida con una cierta frecuencia de muestreo, el método utilizado corresponde a interpolación por *nearest neighbor*. En este caso la nueva frecuencia de muestreo debe ser $f_s^{new} = 256Hz$. Se debe tener la precaución de no violar el criterio de Nyquist (i.e., máxima frecuencia presente en la señal $\leq \frac{f_s^{new}}{2}Hz$).
3. **Normalización *max-min*:** Normaliza cada segmento entre $[-1,1]$. Esto implica que para cada segmento se busca el máximo y mínimo, entre todos los canales del segmento, y luego se calcula cada valor normalizado con la siguiente ecuación

$$x_{ct} = \left(\frac{x_{ct} - \min(X)}{\max(X) - \min(X)} - 0.5 \right) * 2, \quad (3.14)$$

además, se agrega un canal adicional de valor constante en el tiempo, el cual describe la escala relativa del registro EEG con respecto al resto de los registros.

4. **Conservar solo ciertos canales:** Con el interés de minimizar electrodos innecesarios para un problema de alta dimensión, en [60] se preentrenó el modelo solo con 19 canales, *FP1, FP2, F7, F3, FZ, F4, F8, T7, C3, CZ, C4, T8, T5, P3, PZ, P4, T6, O1, O2*, además del canal de escala relativa. Por ende, solo se conservarán los canales recién indicados; y si en el conjunto de datos a usar no se encuentran todos los canales requeridos, entonces los canales faltantes se fijan en 0.

En la figura 3.8 se muestra uno de los segmentos posterior a la aplicación del preprocesamiento indicado, es decir, listo para entrar a la red.

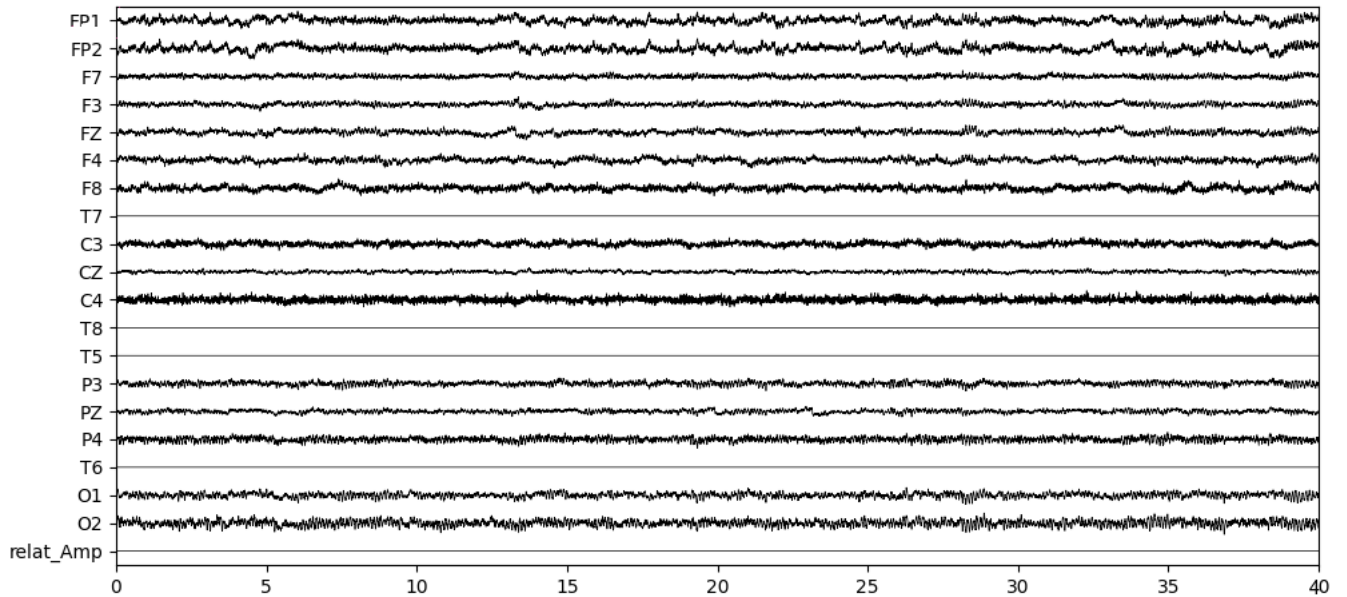


Figura 3.8: Un ejemplo de cómo se ve un segmento justo antes de ingresar a la red. Es decir, un segmento previamente limpiado y preprocesado.

3.4.2. Pesos de preentrenamiento

En [60] se preentrena un modelo completo (conformado por un encoder convolucional y un contextualizador formado principalmente por un *transformer encoder*) utilizando un aprendizaje contrastivo autosupervisado para, principalmente, apuntar a aprender representaciones comprimidas de señales EEG en crudo, bajo ciertos estándares de entrada al modelo. Como datos de preentrenamiento utiliza el conjunto de datos de acceso público “*Temple University Hospital EEG Corpus*” (TUEG) [65], el cual se conforma por registros clínicos usando, en su mayoría, la configuración convencional de registro (electrodos monopolares en configuración 10-20), sobre 10000 sujetos (51% de mujeres, y un rango etario de 1-90 años); lo cual acumula aproximadamente 1.5 *TB*. Posterior al preentrenamiento se propone utilizar *transfer learning*, aprovechando los pesos obtenidos, para afinar el modelo completo según la tarea de clasificación de EEGs deseada.

En el caso de esta tesis, solo se aprovecharán los pesos de la primera parte del modelo, es decir, del encoder convolucional, puesto que la arquitectura que se utiliza, recién descrita, no contará con un contextualizador basado en un *transformer encoder*.

3.4.3. Aumento de datos (*Data augment*)

Se probaron 2 técnicas de aumento de datos. La primera consiste en la recombinación de segmentos de una misma señal para crear señales nuevas, lo cual mejoró los resultados pero no sustancialmente.

La segunda técnica, que debido a sus buenos resultados, se decide utilizar, funciona durante el entrenamiento del modelo y contiene una componente probabilística. El algoritmo del método se describe a continuación en el algoritmo 2.

Algoritmo 2: Algoritmo de *data augmentation* usado durante los entrenamientos.

```
for batch in dataset do
  for sample in batch do
    if prob. 20 % then
      | sample = sample * random[U(0.8, 1.2)] ;      #se modifica amplitud de la señal
    else
      | if prob. 20 % then
      | | sample = sample * (-1) ;                      #se invierte la señal
      | else
      | | sample=sample
      | end
    end
  end
end
end
```

En este algoritmo $random[U(0.8, 1.2)]$ representa un número aleatorio con distribución uniforme entre 0.8 y 1.2. Entonces, en cada iteración del entrenamiento, se recorrerán los segmentos del *batch*, y eventualmente, con una probabilidad de $1 - (0.8 * 0.8) = 0.36$ serán modificados, ya sea modificando su amplitud o invirtiendo la señal entera. En la figura 3.9 se ilustra el método para 1 canal, pero si el segmento se modifica en entonces se modifican

todos los canales por la misma operación. Este método en realidad puede considerarse tanto de *data augmentation* como de regularización.

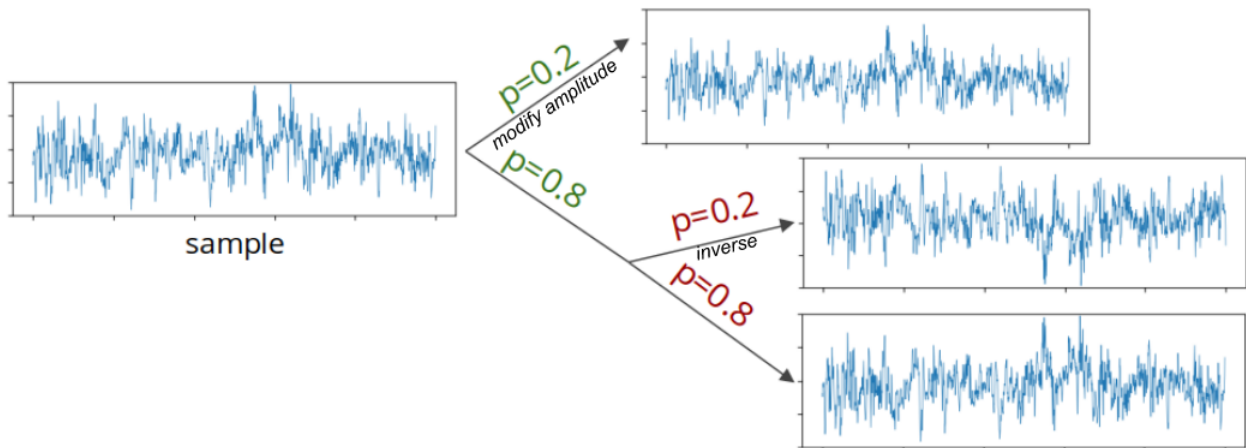


Figura 3.9: Método utilizado en cada *batch*, durante el entrenamiento, para regularizar el entrenamiento.

3.4.4. Evaluación del modelo

Respecto a la manera de evaluar los desempeños de cada modelo, ésta depende de la tarea pedida:

- Datos *EQZvsHC*: Dado que este es un conjunto de datos auxiliar utilizado como primer acercamiento, no se utiliza un conjunto de prueba. Entonces, se entrena y valida en 6 iteraciones de *cross-validation* (CV), donde en cada iteración se retiran 4 registros (2 de cada clase) y se utilizan como conjunto de validación, luego los 8 registros restantes se utilizan para entrenar. Posteriormente se analizan las curvas de entrenamiento generadas por el promedio de las 6 curvas de entrenamiento y validación. Los segmentos de 40 s en este caso son usados tanto para entrenar como para validar, por lo que se consideran como si fuesen señales independientes entre sí.
- Datos *DecompStudy*: Lo primero es escoger sujetos que serán los sujetos de *test*, los cuales no deben estar relacionados con los ajustes del modelo ni con la selección de los hiperparámetros. En este caso se escogen los sujetos SA011 y SA012 (ver tabla 5.1).

Segundo, se debe definir cómo se desea que funcione el clasificador final. De algunas conversaciones con expertos y revisando la bibliografía se tiene que los síntomas de la esquizofrenia no son síntomas que se encuentren presentes a cada momento en un paciente descompensado. Por ende, si un sujeto un día se encuentra con los síntomas más acentuados, esto no necesariamente se reflejará en cada instante del EEG. Es por lo anterior, que a pesar de que el modelo se entrene con segmentos de 40 s, finalmente, se decide que al momento de evaluar no se considerará que cada segmento es independiente, sino que forma parte de un registro completo que es lo que se debe clasificar. Para esto se debe definir algún criterio que explique cómo se va a definir la clasificación de un registro, una vez entrenado el modelo.

El modelo final decide la clasificación final de un registro según el siguiente concepto, y de esta manera también se calcula el *accuracy* de validación:

- Un registro EEG será procesado por el modelo en segmentos de 40 s, con traslape de 30 s. El resultado de clasificación de cada segmento es almacenado.
- Una vez procesados todos los segmentos de un registro, se prosigue a revisar el resultado de cada uno. Aquí se aplica un criterio, definido arbitrariamente y por heurística, que se denomina **detección mínima de “1”s**. Esto es, si existen 3 o más segmentos clasificados como clase “1” (i.e. síntomas positivos elevados), luego el registro completo será clasificado como clase “1”. Esto implica, que para que un registro sea clasificado como “0” (i.e. síntomas positivos regulares/bajos), será necesario que casi todos los segmentos pertenecientes al registro sean clasificados como “0”.
- Lo anterior también implica que si se quiere maximizar el *accuracy*, luego, es necesario minimizar al máximo los falsos positivos, mientras que los falsos negativos no influyen en gran manera por si solos. Esto conlleva a que, como se ve en la sección 3.4.5, se utilice una función de costos ponderada por clases.

La intuición seguida para definir el criterio para calcular el *accuracy* recién mencionado, está basado en la siguiente idea: “un sujeto EQZ con síntomas positivos elevados, no va a presentar necesariamente efectos de esto a todo momento; por otro lado, un sujeto EQZ con síntomas atenuados no debiera casi presentar efectos de los síntomas en ningún momento”.

Por otra parte, tomando en cuenta que la cantidad de registros disponibles por sujeto es limitada, y recordando que el ajuste del modelo se hace con un sujeto a la vez, se tiene que la cantidad de valores posibles que puede tomar el *accuracy* calculado al usar el criterio de detección mínima de “1”s será bastante limitado. Dado lo anterior, para efectos de calcular las *accuracies* de validación durante el entrenamiento, se divide cada registro en 3 partes, que son tratadas como si fueran 3 subregistros diferentes (ver figura 3.10). Aunque por ser subregistros más pequeños se define, por heurística también, que la mínima cantidad de segmentos clasificados “1” que se requerirán serán 2 (en vez de 3). De esta manera, las curvas de validación presentan una forma más fluida y entregan más información acerca de lo que está ocurriendo en el entrenamiento. Por el contrario, las curvas con el conjunto de entrenamiento tratan cada segmento de 40 s como si fuese un registro independiente.

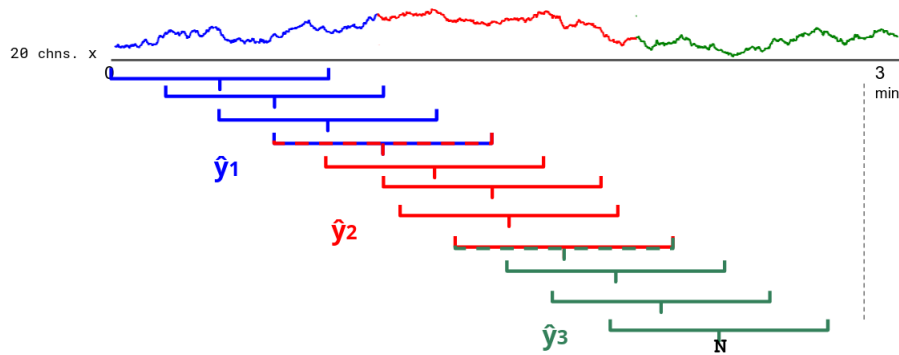


Figura 3.10: Ilustración del criterio utilizado para computar los valores de las curvas de validación. Se ilustra con un canal, pero la misma idea se extiende a los 20 canales que se usan.

Cabe mencionar también que se entrenará con validación cruzada (CV, por sus siglas en inglés), donde el número de iteraciones a realizar dependerá del sujeto que se esté usando para ajustar el modelo. Para cada iteración de CV, deben quedar 2 registros completos afuera del entrenamiento, y uno con etiqueta “1” y el otro con etiqueta “0”. La cantidad de iteraciones de CV por sujeto, se pueden revisar en la tabla 5.1.

3.4.5. Hiperparámetros del modelo

La selección de hiperparámetros se realizó a través de una búsqueda aleatoria, donde se muestrearon diferentes valores para cada caso. Los hiperparámetros seleccionados para cada conjunto de datos entrenado se presentan en la tabla 3.2.

Tabla 3.2: Hiperparámetros para el conjunto de datos *EQZvsHC*.

Hiperparámetro	<i>EQZvsHC</i>	<i>DecompStudy</i>
Función de costo	Entropía cruzada	Entropía cruzada pond. [1, 0.2]
Tasa de aprendizaje	$1e - 4$	$1e - 4$
Regularización L2 β (<i>weight decay</i>)	0.01	0.01
Tamaño de <i>batch</i>	8	8
Criterio de optimización	Adam	Adam
Tasa dropout en codificador	0.3	0.3
Tasa dropout en decodificador	0.7	0.7
Umbral de clasificación ^a	0.4	0.4

^a Valor que debe sobrepasar la probabilidad de una salida del modelo, de clasificación binaria, para ser clasificado como clase “1”. La probabilidad de una salida se calcula tomando la sigmoide del *logit* o puntaje final que entrega el modelo.

Como se observa los hiperparámetros en su mayoría son los mismos, excepto por la función de costo. La función de costo ponderada para el conjunto *DecompStudy* no se escoge por desbalance de clases, sino que por lo mencionado en la sección 3.4.4 acerca de la evaluación del modelo. Dado que la clasificación de un registro como clase “1” (i.e. síntomas positivos elevados) será dada por la detección de un mínimo pequeño de segmentos dentro de un registro completo, entonces, resulta que el costo de un falso negativo (un segmento “1” que se predice

como “0”) debiera ser bastante bajo, en comparación a la clasificación de los registros de clase “0”, que debieran presentar todos sus segmentos con predicción “0”; luego, el costo de un falso positivo (un segmento “0” que se predice como “1”) debiera ser alto. Es por lo anterior que la entropía cruzada ponderada va más de acuerdo con maximizar el *accuracy* para el conjunto de datos *DecompStudy*.

3.5. Softwares utilizados

Los principales softwares utilizados para la metodología de estas tesis son los siguientes:

- **MATLAB**[®][66]: Plataforma de programación y computación numérica. Se usó principalmente, junto a la *toolbox* EEGLAB[67], para el manejo y preprocesamiento de las señales EEG. La *Deep Learning toolbox* también fue de gran ayuda, pues se utilizó para una etapa crucial del preprocesamiento de las señales.
- **Python**[™][68]: Lenguaje de programación utilizado para la obtención de *features* de las señales, así como también el postprocesamiento de éstas. Particularmente, se utilizó la librería Pytorch v1.10 para el desarrollo y ejecución de los modelos de *machine learning*.

4. Resultados

4.1. Enfoque con extracción manual de características

Entre las soluciones entregadas por el método de selección de características basado en Información Mutua, y ayudándose, además, por modelos de *random forest* (RF) sobre todas las características, se escogieron finalmente los siguientes 3 conjuntos de características que estarían relacionados con el nivel de síntomas positivos de cada paciente:

- [IQR de banda beta, Potencia promedio de banda theta] (figuras 4.1, 4.2, 4.3).
- [STD de banda alfa, LZC-Hilbert de banda completa (0.1-100 Hz), MAD de banda delta, Potencia promedio de banda theta] (figuras 4.4, 4.5, 4.6).
- [STD de banda alfa, LZC-Hilbert-pow. de banda completa (0.1-100 Hz), IQR de banda Delta, STD de banda theta] (figuras 4.7, 4.8, 4.9).

Estos conjuntos fueron evaluados visualmente, y resultan ser una buena primera aproximación a la relación entre los registros EEG y el estado sintomático de los pacientes con EQZ. Los 3 conjuntos arriba mencionados fueron escogidos evaluando los gráficos de dispersión de los segmentos de registros de los sujetos SA007, SA010, SA014, SA025, SA026, SA039 y SA047; dejando reservados los segmentos de SA011 y SA012, para evaluarlos al final una vez ya decididos los conjuntos de características a emplear. A continuación se presentan los gráficos de dispersión (figuras 4.1-4.9).

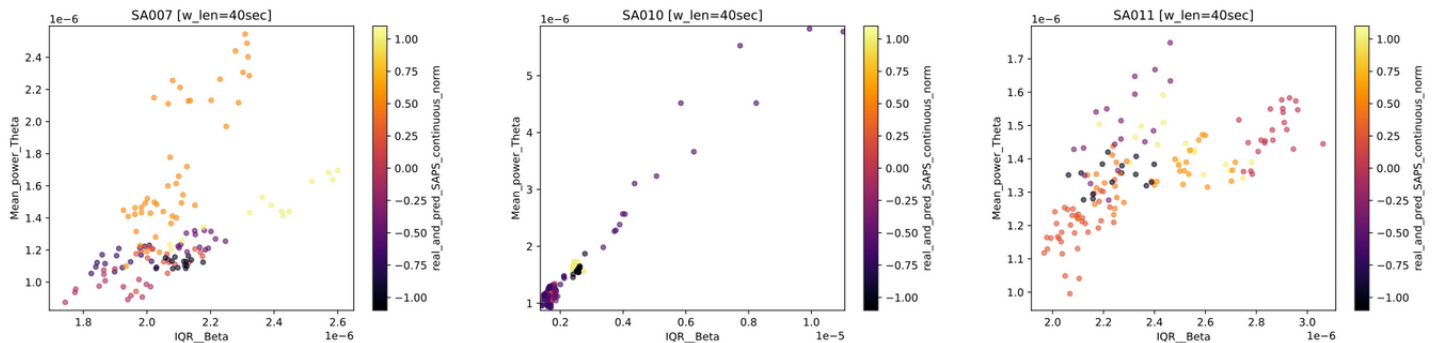


Figura 4.1: Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.

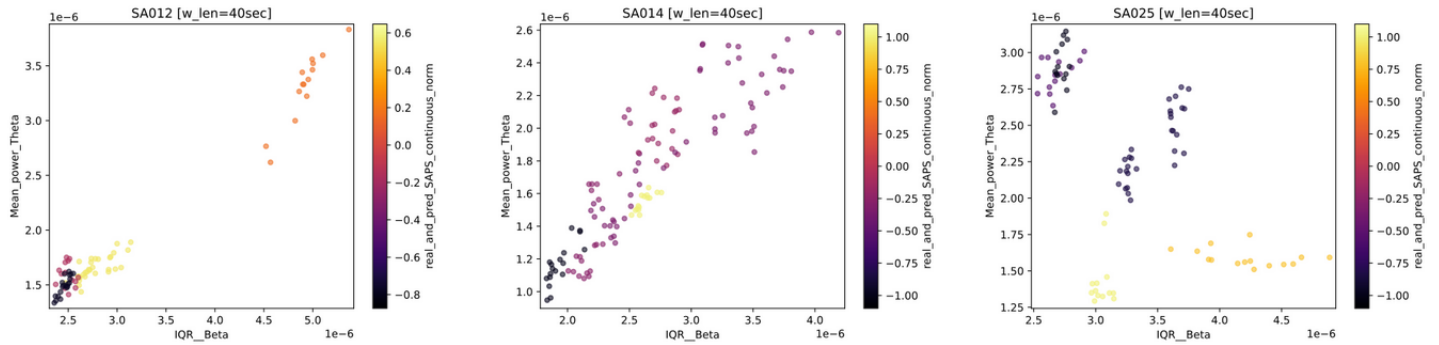


Figura 4.2: Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.

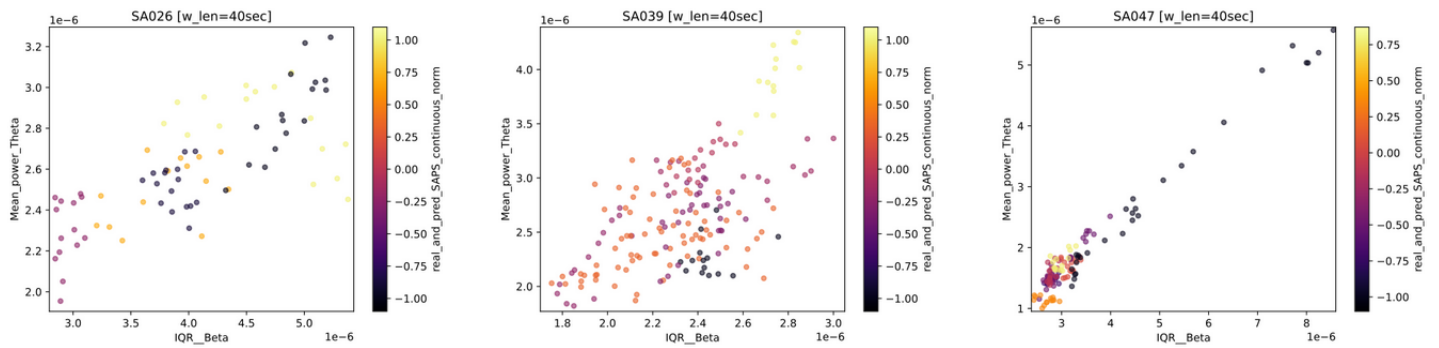


Figura 4.3: Gráfico de dispersión de características ['IQR_Beta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.

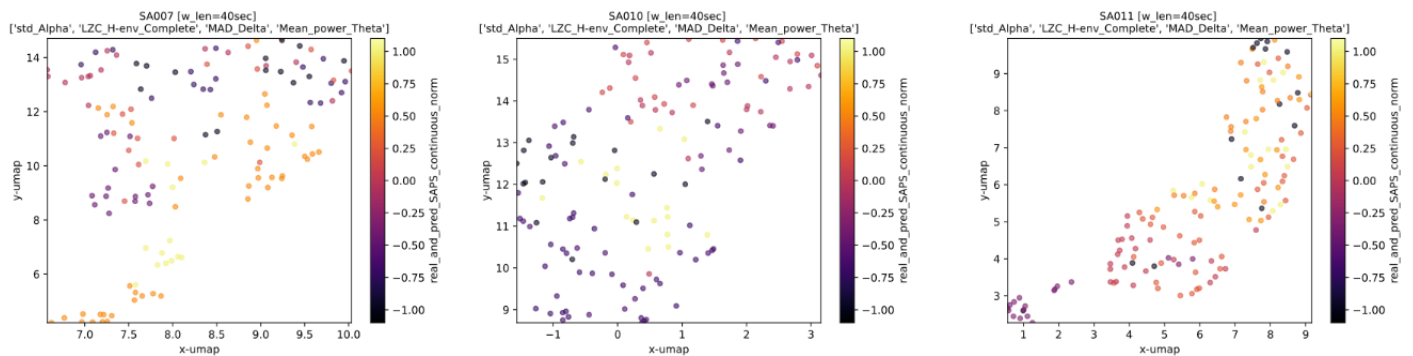


Figura 4.4: Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.

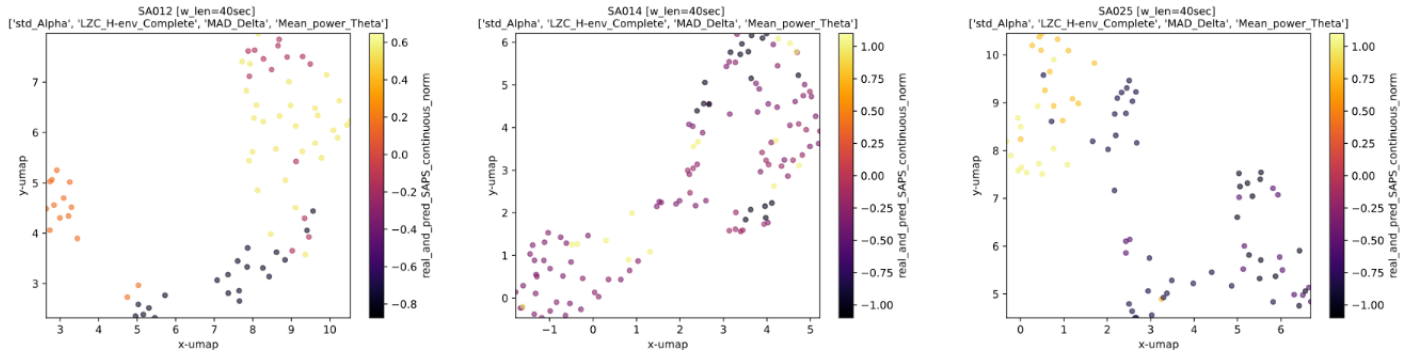


Figura 4.5: Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.

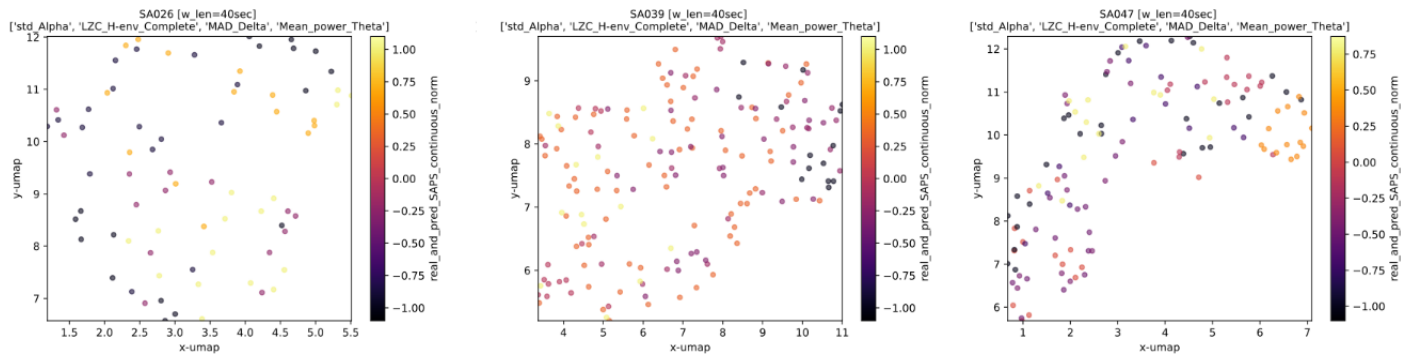


Figura 4.6: Gráfico de dispersión de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.

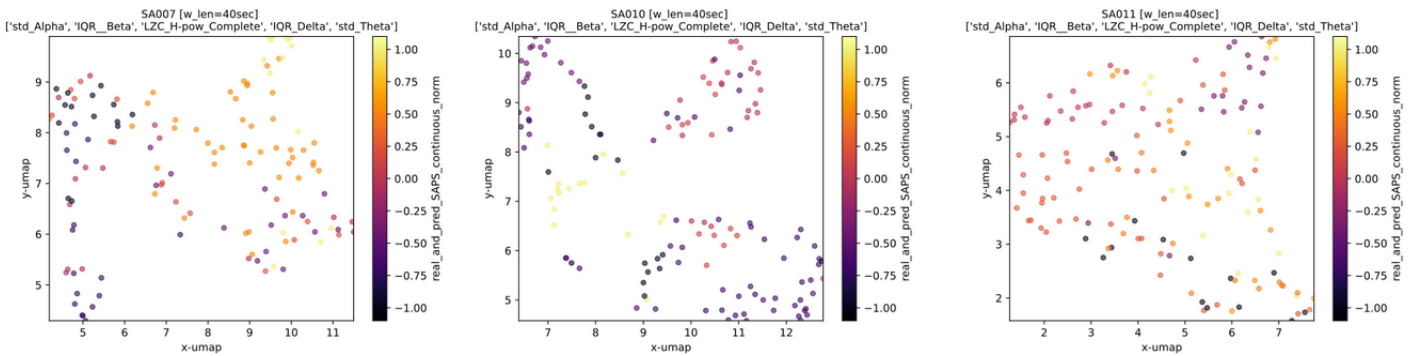


Figura 4.7: Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA007, SA010 y SA011. Los colores indican la etiqueta continua normalizada por sujeto.

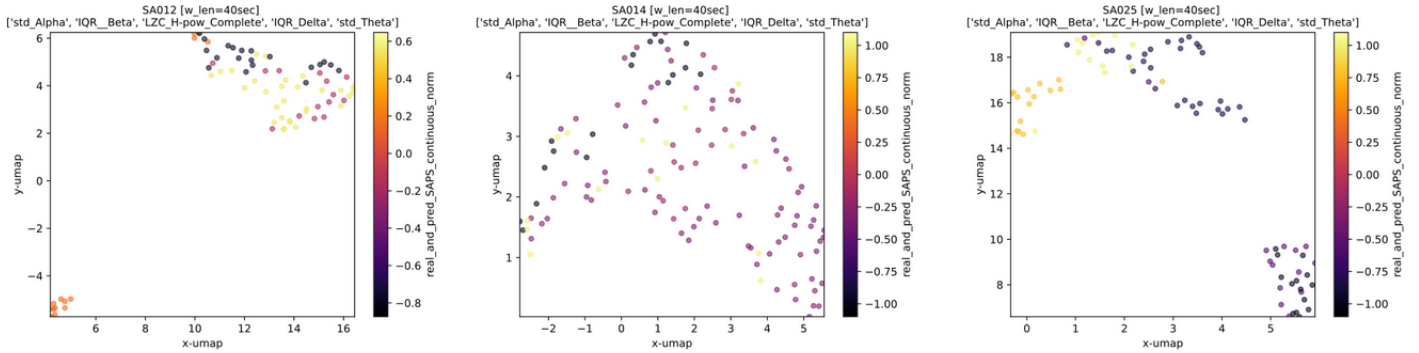


Figura 4.8: Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA012, SA014 y SA025. Los colores indican la etiqueta continua normalizada por sujeto.

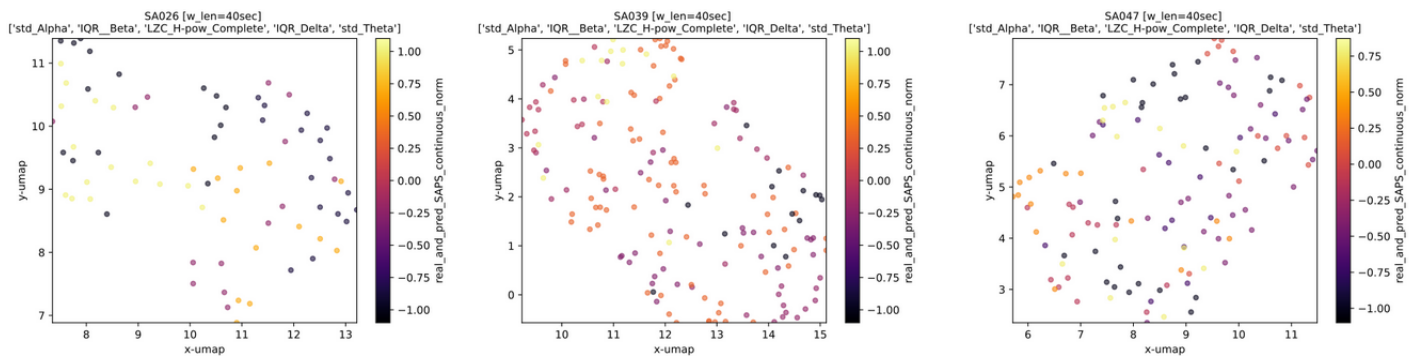


Figura 4.9: Gráfico de dispersión de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] de cada segmento de 40 s, para los sujetos SA026, SA039 y SA047. Los colores indican la etiqueta continua normalizada por sujeto.

Posterior a la evaluación visual de los 3 conjuntos de características propuestos con respecto a la relación que estos podrían tener con el estado de los síntomas positivos de cada paciente, se procede a analizar esta misma posible relación pero por medio de modelos RF. Para cada sujeto se entrena un modelo de RF para cada conjunto de características, las cuales son entregadas como entrada para cada modelo. Para cada sujeto, el 25 % de los segmentos de 40 s son reservados para probar la utilidad de los modelos.

Los resultados obtenidos con RF utilizando el primer conjunto de características fueron omitidos debido a su mal rendimiento. Por otro lado, los resultados obtenidos con los otros dos conjuntos de características se presentan en las tablas 4.1 y 4.2; donde R^2 corresponde a el coeficiente de determinación de los modelos, definido como $R^2 = (1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\text{mean}(y))^2})$, y el cual representa la proporción de la varianza total de la variable *target* explicada por el modelo de regresión. Por otro lado, en las tablas se destaca para cada modelo la característica que presentó una mayor importancia de permutación⁷ con respecto al resto, siempre y cuando fuera una importancia considerable.

⁷ Mide cuánto aumenta el error de predicción del modelo después de permutar los valores de la característica, lo cual rompe la relación entre la característica y el resultado real.

Tabla 4.1: Resultados de *random forest* utilizando el conjunto de características ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'] como entrada.

Sujeto	R^2 (<i>train</i>)	R^2 (<i>test</i>)	<i>important feature</i>
SA007	0.8754	0.7311	LZC_H-env_Complete
SA010	0.8321	0.6250	MAD_Delta
SA011	0.6485	0.5129	LZC_H-env_Complete
SA012	0.9219	0.9591	MAD_Delta
SA014	0.7763	0.6514	MAD_Delta
SA025	0.9901	0.9567	Mean_power_Theta
SA026	0.8817	0.5272	LZC_H-env_Complete
SA039	0.7326	0.7419	LZC_H-env_Complete
SA047	0.6457	0.4607	LZC_H-env_Complete

Tabla 4.2: Resultados de *random forest* utilizando el conjunto de características ['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta'] como entrada.

Sujeto	R^2 (<i>train</i>)	R^2 (<i>test</i>)	<i>important feature</i>
SA007	0.8637	0.7866	LZC_H-pow_Complete
SA010	0.8360	0.7534	IQR_Delta
SA011	0.6880	0.4654	<i>inconclusive</i>
SA012	0.9855	0.9322	IQR_Delta
SA014	0.9326	0.7586	IQR_Delta
SA025	0.9625	0.9230	std_Theta
SA026	0.8327	0.7657	LZC_H-pow_Complete
SA039	0.8158	0.6525	LZC_H-pow_Complete
SA047	0.6690	0.4741	IQR_Beta

Cabe mencionar que los modelos RF fueron definidos con 30 estimadores (i.e. 30 árboles de decisión por modelo), se utilizó MSE como criterio de entrenamiento, y se definió una profundidad máxima de 4. Ya que para cada modelo se ajustaron 30 árboles de decisión, no es posible entregar una visualización que de los modelos que sea lo suficientemente concisa y explicativa; sin embargo, de manera ilustrativa, se muestra en la figura 4.10 cómo luce uno de los árboles de decisión ajustado con los datos del sujeto SA007 para el conjunto ['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta'].

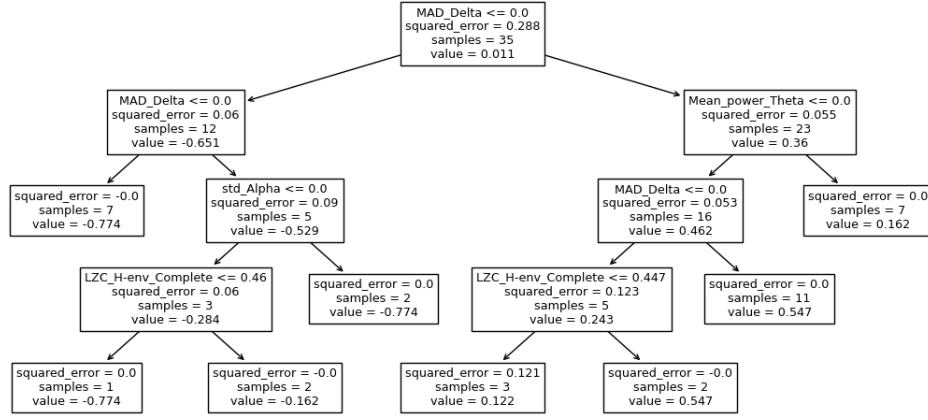


Figura 4.10: Muestra de uno de los árboles de decisión resultante para el sujeto SA007.

4.2. Enfoque con aprendizaje profundo

4.2.1. Primera aproximación con datos auxiliares

Al igual que con el enfoque de extracción manual de características, antes de utilizar los datos de *DecompStudy*, como *sanity check* se prueba el enfoque con el conjunto de datos *EQZvsHC*. Cabe recordar, que antes de la utilizar estos datos, el conjunto *EQZvsHC* es procesado por la pipeline de limpieza de señales descrito en la sección 3.2.1.

Con el propósito de evaluar la capacidad de representación del codificador del modelo preentrenado en [60], se representa gráficamente el estado intermedio de las señales dentro del modelo, z , i.e. justo después de pasar por el codificador, para ver si existe algún tipo de *clusterización* de los segmentos según su etiqueta. Dado que la señal en estado latente, z , posee una dimensión de $(512, 107)$, i.e. 512 canales de largo 107 *features* cada uno, se debe definir alguna manera para poder ser representada en 2D. Para ello se utiliza el siguiente método:

1. Se crea un nuevo vector de largo 107, donde se almacena el índice del canal que presenta mayor activación en cada *feature*.
2. En otro vector de largo 107, se almacena el promedio de todos los canales para cada *feature*.
3. Se concatenan ambos vectores, dando como resultado un vector 1D de largo 214. Es decir, por cada segmento de 40 s , se tiene una representación en espacio latente de 214 *features*. Esto, para N segmentos, nos da un tensor final de dimensión $(N, 214)$.
4. Finalmente, se reduce el espacio de $(N, 214)$ a $(N, 2)$, por medio del algoritmo *Uniform Manifold Approximation and Projection for Dimension Reduction* (UMAP), el cual es capaz de reducir dimensionalidad utilizando estructuras topológicas difusas [59]. Se utilizan los mismos parámetros de UMAP usados al analizar las características en la sección 3.3.2 (i.e. tamaño de vecindad 5 y distancia de Mahalanobis).

De esta manera, realizando el procedimiento anterior y reduciendo a solo 2 “*features*”, se obtienen los gráficos de dispersión presentados en la figura 4.11. De ellos se puede observar

que el codificador preentrenado, sin siquiera haber visto antes los datos de *EQZvsHC*, logra formar clústers por etiqueta EQZ vs HC.

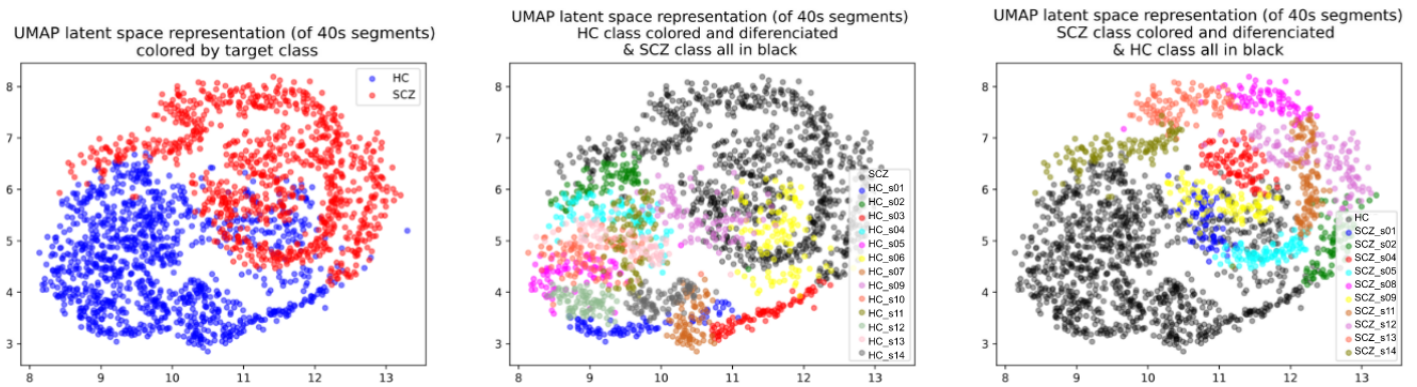


Figura 4.11: Proyección 2D de la representación de segmentos de 40 s en el espacio latente obtenida por el codificador convolucional, para la base de datos *EQZvsHC*.

Una vez validado el potencial del codificador preentrenado, se pretende volver a reafirmar su utilidad. Esta vez aplicando transferencia de aprendizaje.

Se entrena la arquitectura con CV sobre los registros, los cuales luego son segmentados en ventanas de 40 s y tratados como segmentos independientes.

Para analizar el potencial del codificador preentrenado, primero se realiza el entrenamiento con las capas inicializadas por los métodos por defecto en Pytorch, es decir sin utilizar transferencia de aprendizaje. Luego, se realiza el entrenamiento inicializando las capas del *encoder* con los pesos de [60]. Las curvas de generadas para ambos tipos de entrenamiento se ilustran en la figura 4.12.

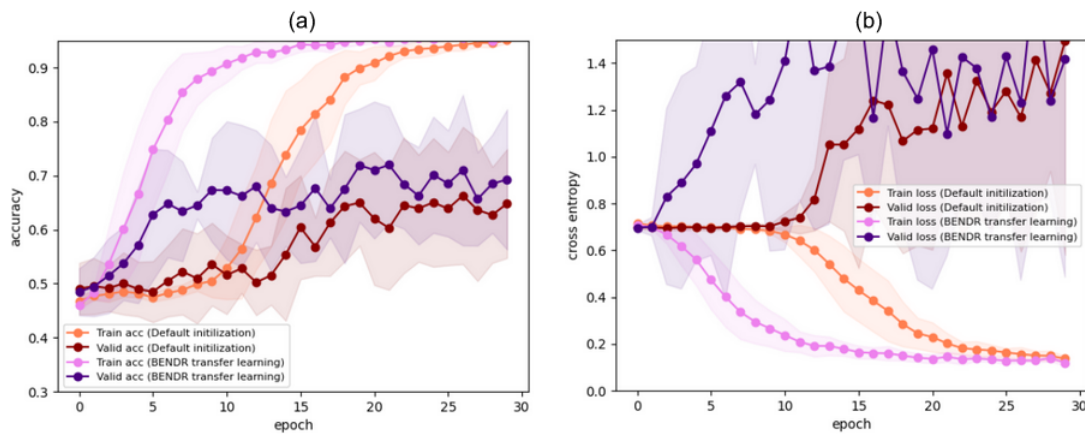


Figura 4.12: Comparación de las curvas de entrenamiento/validación al modificar la inicialización de los pesos, entrenando con los datos *EQZvsHC*. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Curvas rojo-naranja corresponden a una inicialización por defecto, mientras que las curvas rosa-morado corresponde a la utilización de transferencia de aprendizaje.

En la figura 4.12 se observa que al utilizar transferencia de aprendizaje, el desempeño del modelo aumenta rápidamente, y el *accuracy* de validación alcanza un valor mayor en compa-

ración a no utilizar transferencia de aprendizaje. Por otro lado, se observa un comportamiento extraño en las curvas de pérdida (entropía cruzada) en validación, pues estas no decrecen y, por el contrario, terminan con un comportamiento creciente; esto último será mayormente comentado en la sección 5 de Discusión. En lo que sigue, se realizan experimentos siempre usando transferencia de aprendizaje.

Luego, se experimenta con el congelamiento de pesos. Se prueba congelar todo el codificador durante los entrenamientos, así como también congelar solo algunas capas. Los resultados congelando el codificador completo, así como congelando las primeras 3 capas resultan en un modelo infra-ajustado (i.e., presenta *underfitting*), pues la capacidad de ajuste disponible no es capaz de aprender la tarea deseada. Por otro lado, los resultados al congelar solamente la primera capa del codificador si mejoran el desempeño del entrenamiento, tal como se observa en la figura 4.13.

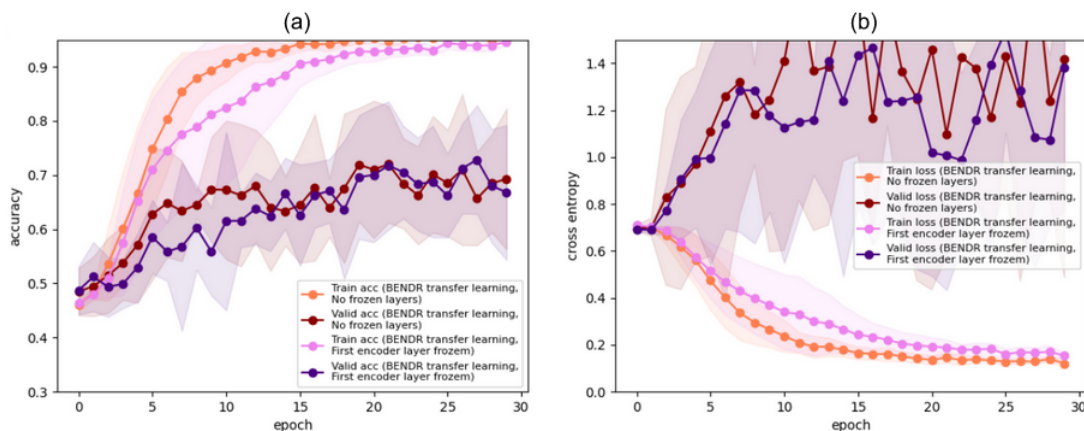


Figura 4.13: Comparación de las curvas de entrenamiento/validación sin congelar parámetros (curvas rojo-naranja) y con congelamiento del inicio de la red (rojas rosa-morado), entrenando con los datos *EQZvsHC*. Curvas de *accuracy* en (a) y curvas de *loss* en (b).

A pesar de que aun persiste el problema de la curva de pérdida creciente en validación, se observa que al congelar la primera capa del codificador el modelo tiende a converger más rápido, y de hecho alcanza valores de *loss* un poco más bajos en comparación a no congelar ninguna capa.

4.2.2. Resultados con base de datos *DecompStudy*

En esta sección, se realizan experimentos con la base de datos *DecompStudy*. Por lo visto en la sección anterior, los pesos de [60] presentan un buen potencial. Se entrena el modelo utilizando desde un principio transferencia de aprendizaje, y congelando la primera capa del codificador. Pero en esta ocasión, se cuenta con 2 conjuntos de pesos que pueden ser de ayuda para la tarea requerida; esto pues se cuenta con los pesos de [60]; y además, se cuenta con los pesos recién obtenidos al entrenar con el conjunto de datos auxiliar *EQZvsHC*, los cuales están ajustados para diferenciar entre EQZ y HC, lo cual podría ser ventajoso para la tarea de discriminación de estados sintomáticos.

En este caso se utiliza el criterio de detección mínima de “1”s, con la configuración descrita en la sección 3.4.5, es decir, se utiliza una función de costos ponderada, la cual está dirigida a

incrementar el *accuracy* como se definió (ver criterio de detección mínima de “1”s, en sección 3.4.4).

Las curvas de aprendizaje utilizando los diferentes pesos de inicialización, usando los segmentos de cada sujeto con CV, se muestran en las figuras 4.13-4.19.

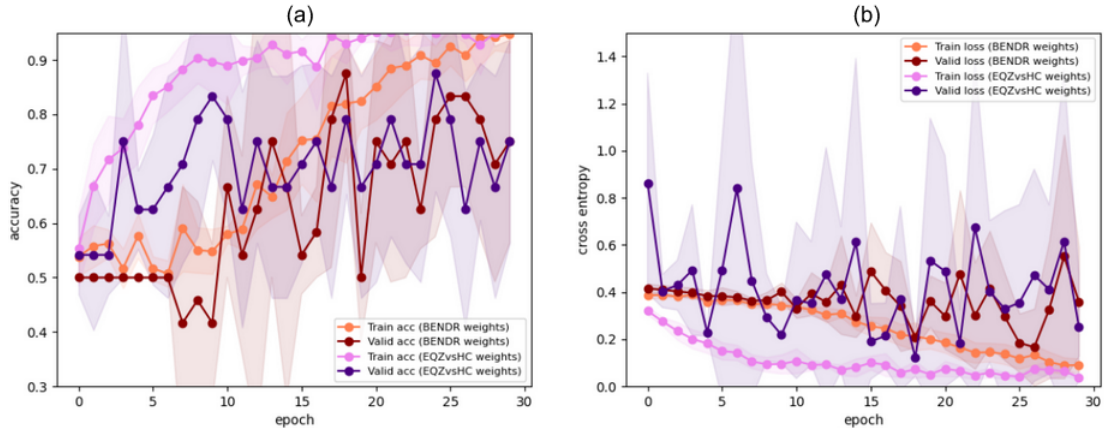


Figura 4.14: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA007** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

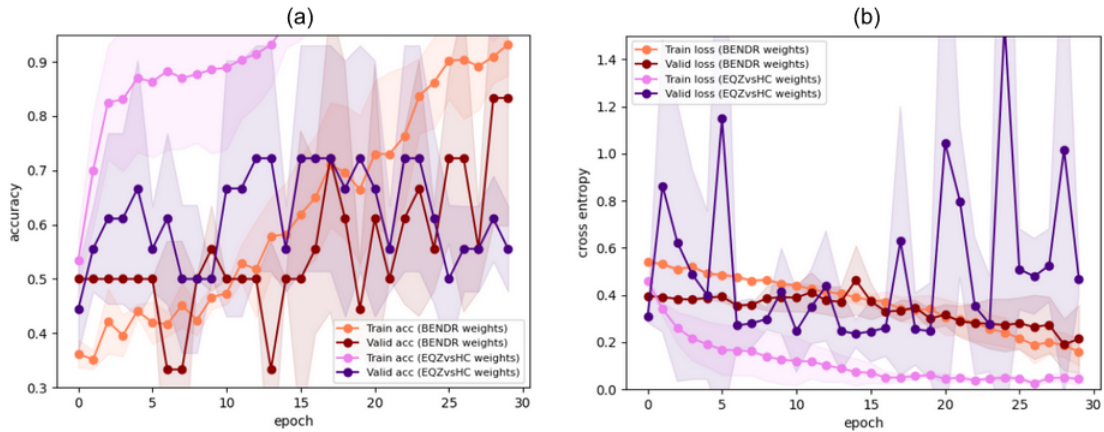


Figura 4.15: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA010** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

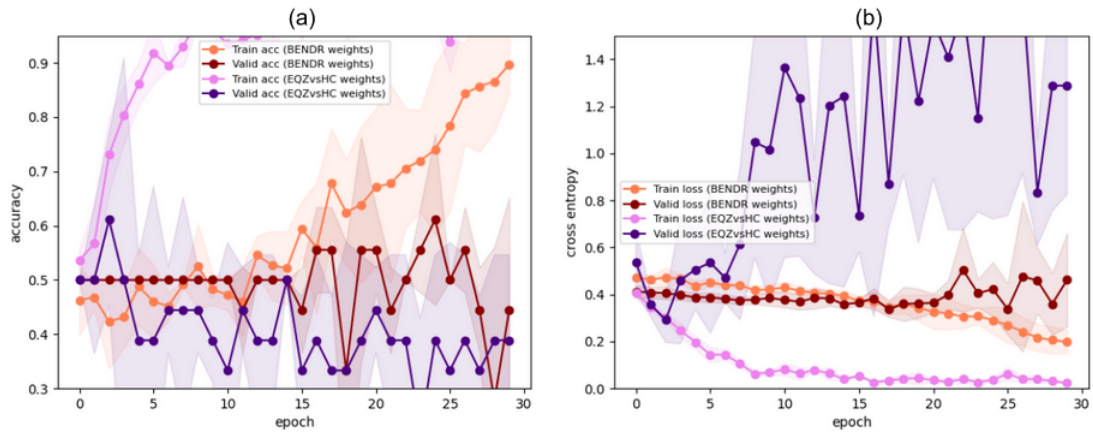


Figura 4.16: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA014** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

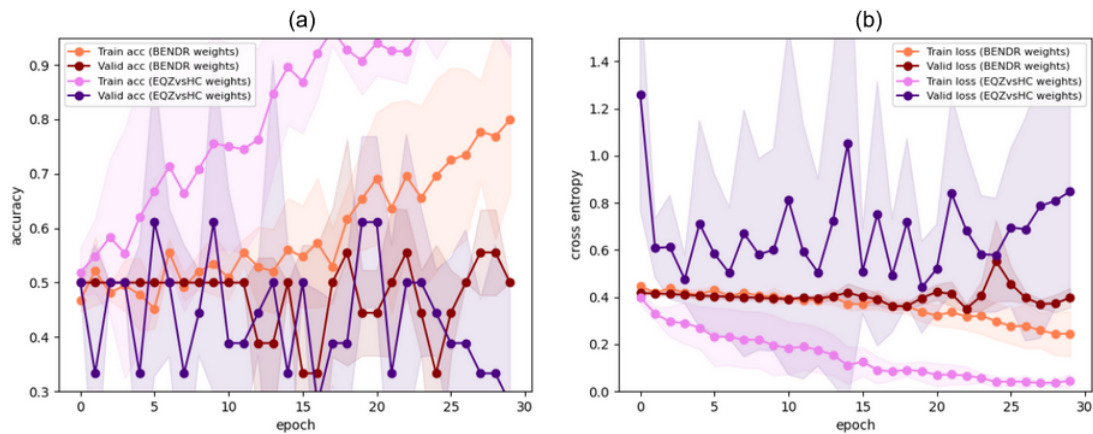


Figura 4.17: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA025** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

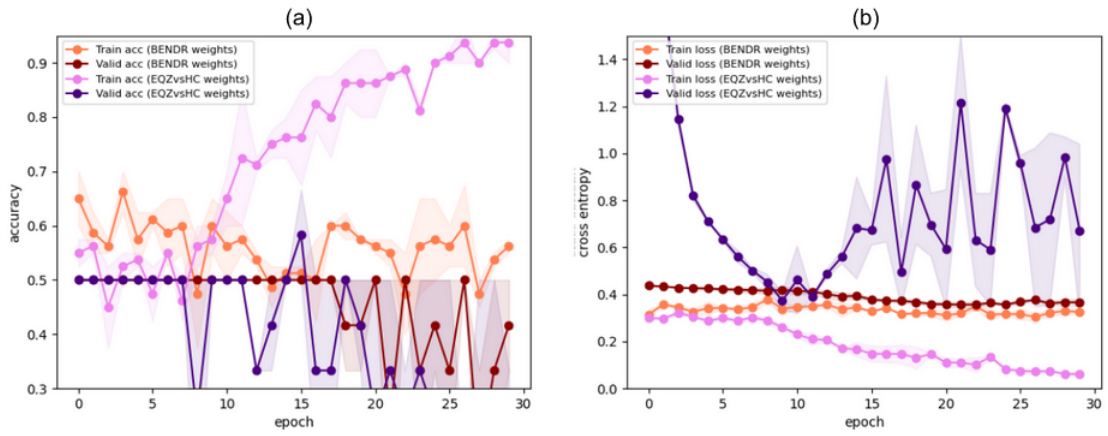


Figura 4.18: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA026** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

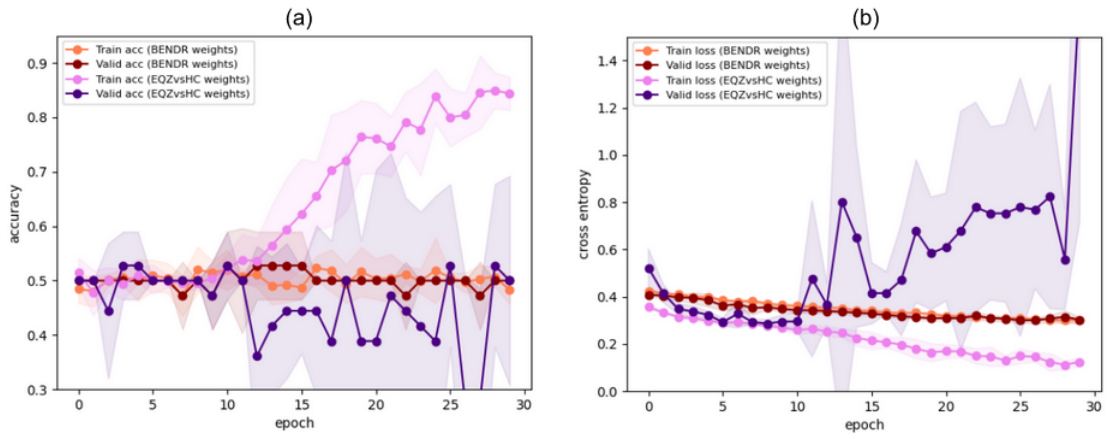


Figura 4.19: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA039** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

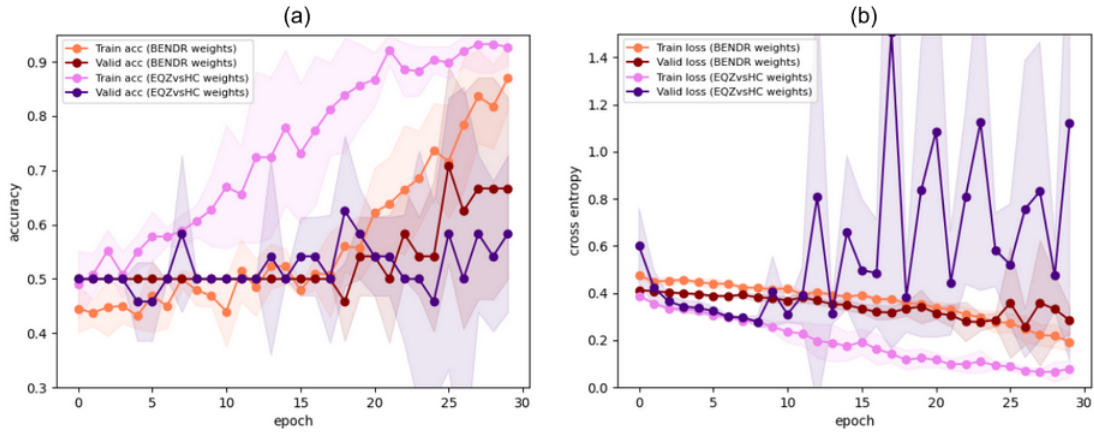


Figura 4.20: Comparación de curvas de entrenamiento y validación al entrenar con el sujeto **SA047** con diferentes inicializaciones de parámetros. Curvas de *accuracy* en (a) y curvas de *loss* en (b). Dentro de cada gráfico, las curvas rojo-naranja corresponden a la utilización de los pesos de preentrenamiento de [60], mientras que las curvas rosa-morado corresponden a la utilización de los pesos obtenidos en la sección 4.2.1.

Para comparar los resultados obtenidos con ambos entrenamientos para cada sujeto, se elige como **modelo resultante** de cada entrenamiento (y para cada iteración de CV), el modelo correspondiente a la época en que el *accuracy* de validación alcanza un mayor valor.

Para cada entrenamiento existe un modelo resultante por cada iteración de CV, para cada sujeto. Entonces, se procede a evaluar cada uno de los modelos resultantes de cada iteración de CV, con los registros de validación correspondientes a esa iteración. Esto se realiza mostrando gráficamente la predicción de cada segmento de los registros completos de validación, disponiendo los segmentos en el orden temporal que corresponden. **En cada gráfica** se representan las predicciones de los segmentos que corresponden al mismo registro, es decir, **a un mismo día** (representando una semana específica); así, se muestra las predicciones de los segmentos de validación para ambos tipos de entrenamiento (entrenamiento con pesos de [60] y entrenamiento con pesos de *EQZvsHC*). La etiqueta real de cada registro se indica en el título de cada gráfico (*RealLabel*), recordando que *RealLabel*=0 indica que el registro fue obtenido cuando el sujeto presentó síntomas positivos bajos/moderados, y que *RealLabel*=1 indica que el registro fue obtenido cuando el sujeto presentó síntomas positivos elevados. Por otro lado, los puntos en los gráficos indican la predicción de cada modelo para cada segmento, mientras que las curvas punteadas indican la probabilidad de salida que arroja el modelo. Recordar que se utiliza un umbral de clasificación de 0.4. A continuación se presentan los gráficos (figuras 4.20-4.26), donde el color azul representa el modelo obtenido con la inicialización de [60], y el color anaranjado representa el modelo obtenido con la inicialización de los pesos de *EQZvsHC*; notar que cuando hay dos puntos sobrepuestos, estos se representan con un tercer color más rojizo.

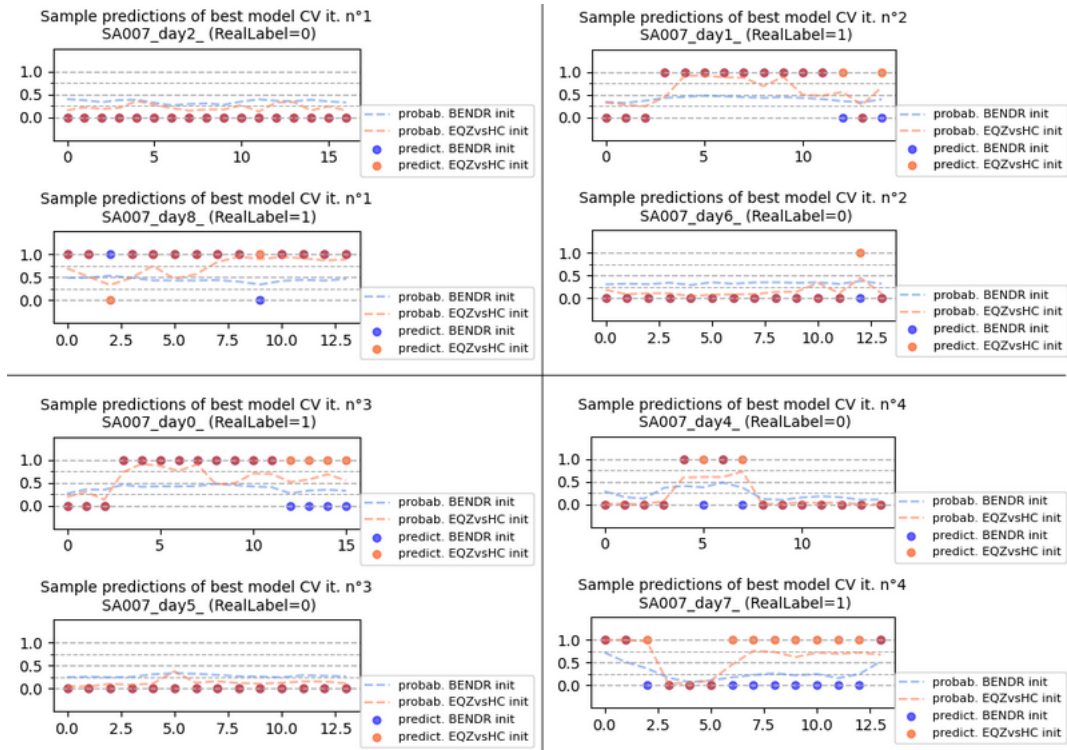


Figura 4.21: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA007**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

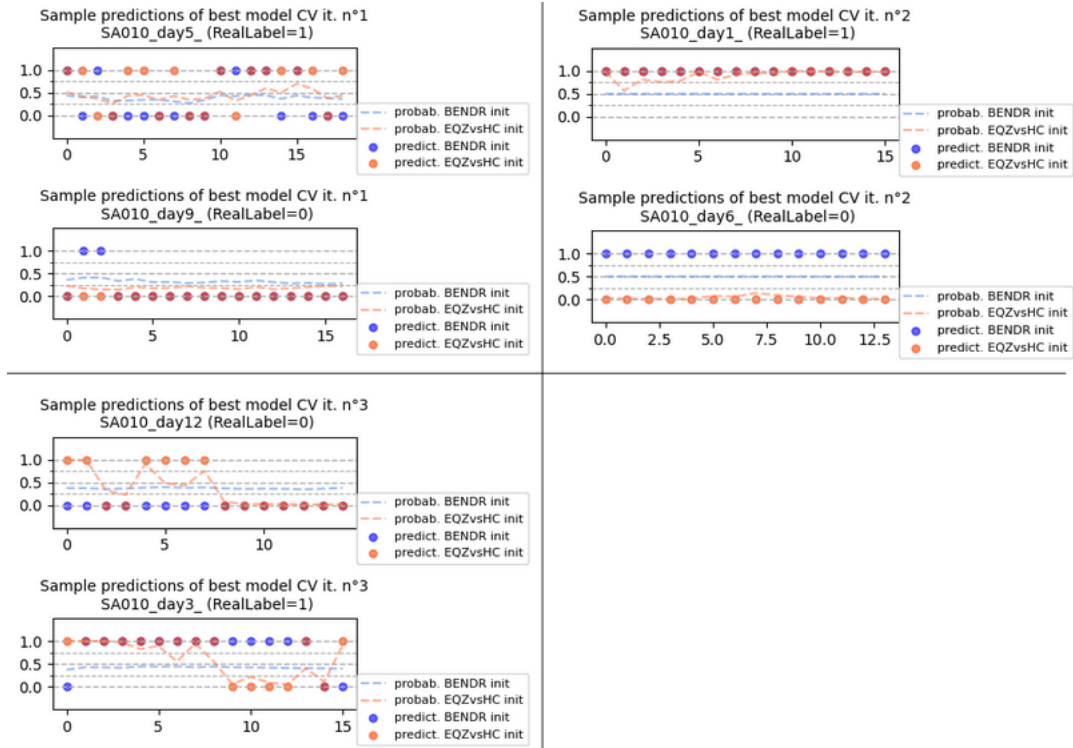


Figura 4.22: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA010**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

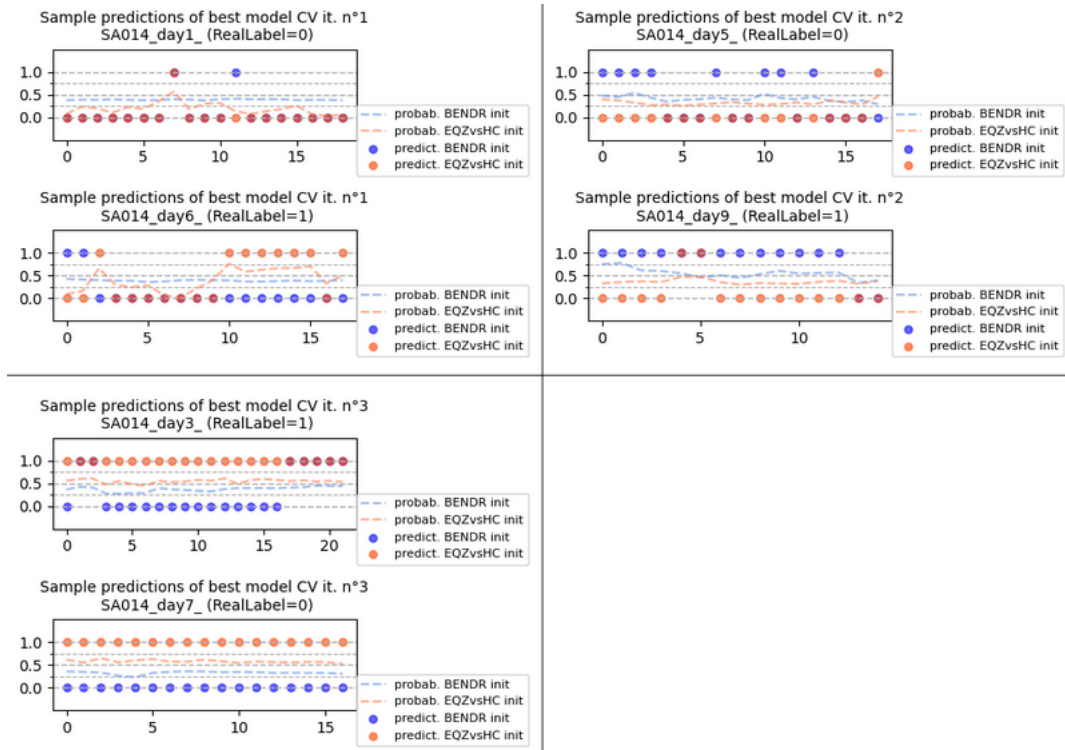


Figura 4.23: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA014**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

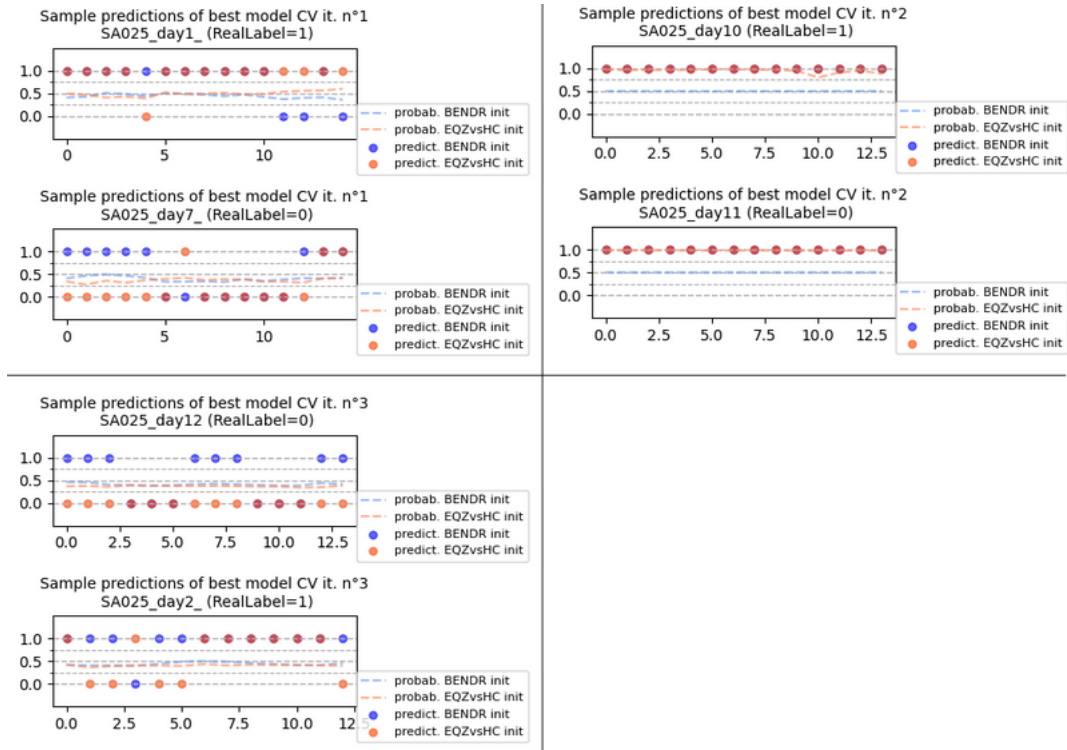


Figura 4.24: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA025**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

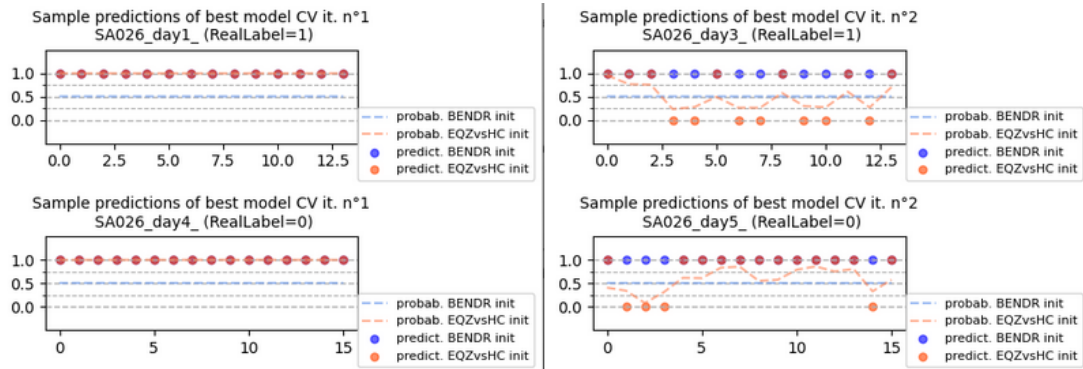


Figura 4.25: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA026**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

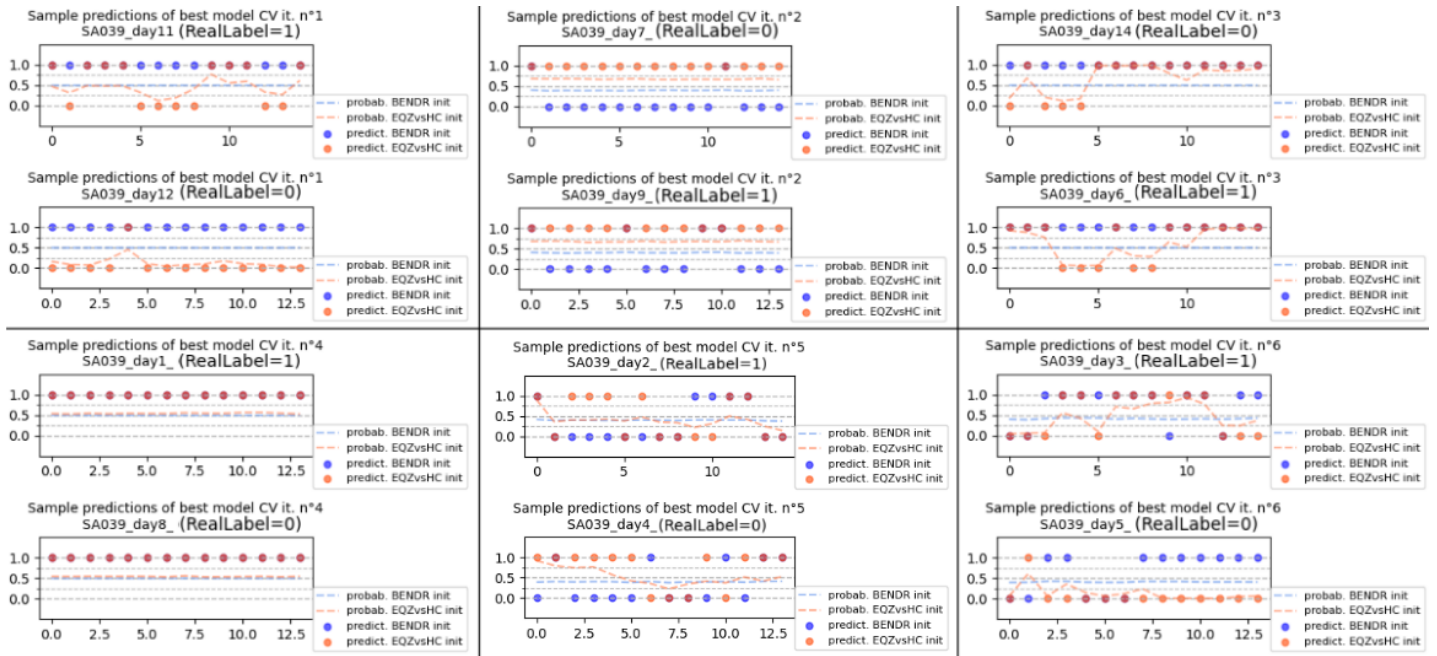


Figura 4.26: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA039**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

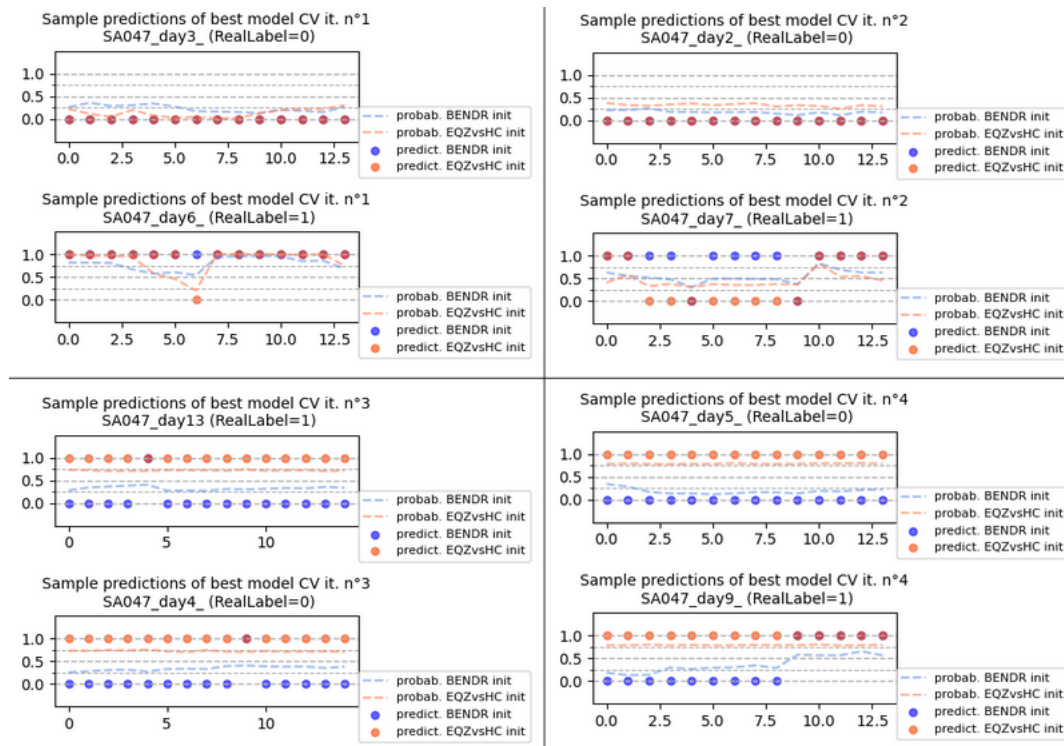


Figura 4.27: Predicciones de los modelos entrenados con pesos iniciales de [60] vs. pesos de *EQZvsHC*, para registros EEG del sujeto **SA047**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

Finalmente, por las razones explicadas en el capítulo de Discusión, se decide que la inicialización por pesos obtenidos en el entrenamiento de *EQZvsHC* resulta más efectiva. Se prueba esta configuración en los registros de los sujetos de *test*, que no se han utilizado para realizar ningún tipo de ajuste de hiperparámetros. Lamentablemente, por la naturaleza del problema, por la poca cantidad de registros disponibles, y por la decisión de ajustar el modelo por sujeto, es que no se puede realizar una prueba más “limpia” en las condiciones actuales, es decir, el modelo debe ajustarse previamente a los sujetos de prueba para ser probado. En las figuras 4.27-4.30 se presentan las curvas de aprendizaje y su posterior gráfico de predicciones, para los sujetos SA011 y SA012, los cuales serán evaluados con CV.

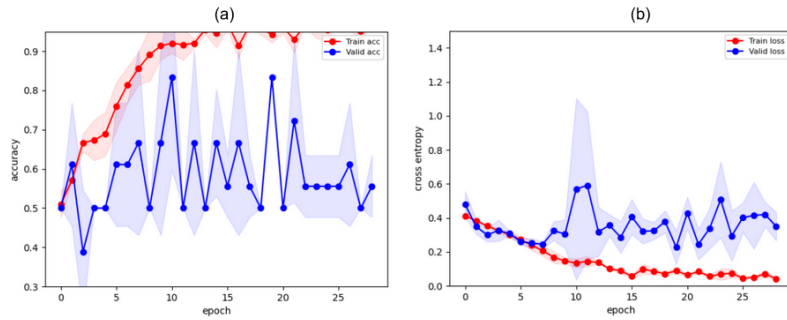


Figura 4.28: Curva de entrenamiento y validación al entrenar con el sujeto **SA011**. Curvas de *accuracy* en (a) y curvas de *loss* en (b).

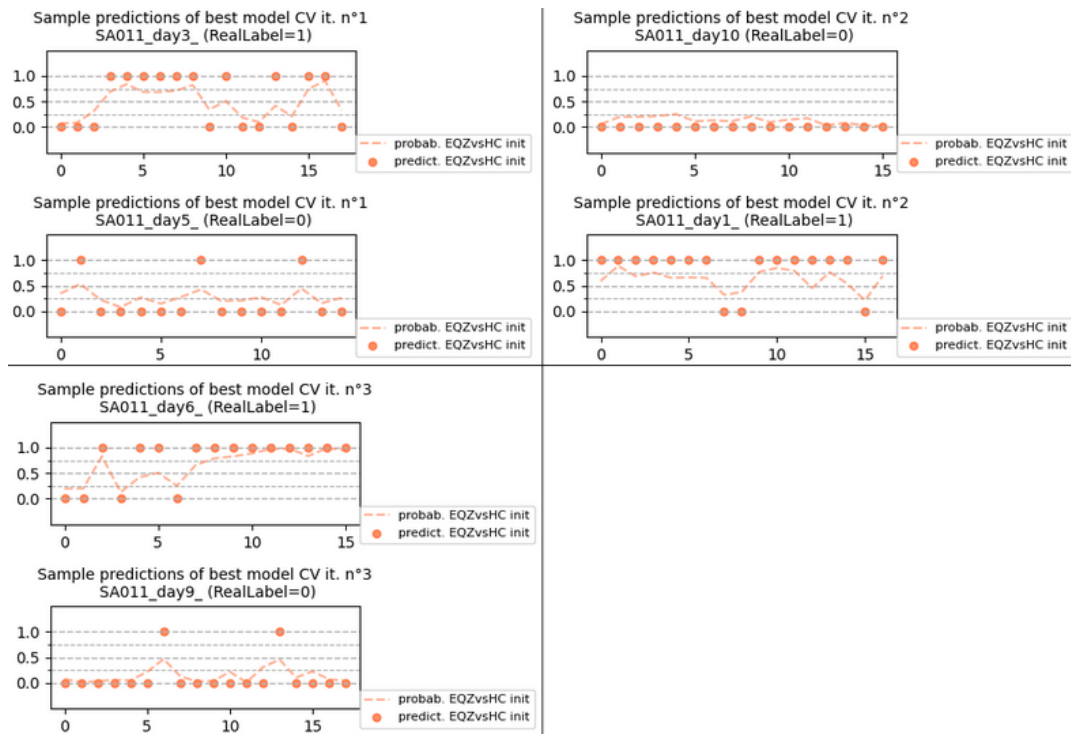


Figura 4.29: Predicciones del modelo entrenado con pesos iniciales de *EQZvsHC*, para registros EEG del sujeto **SA011**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

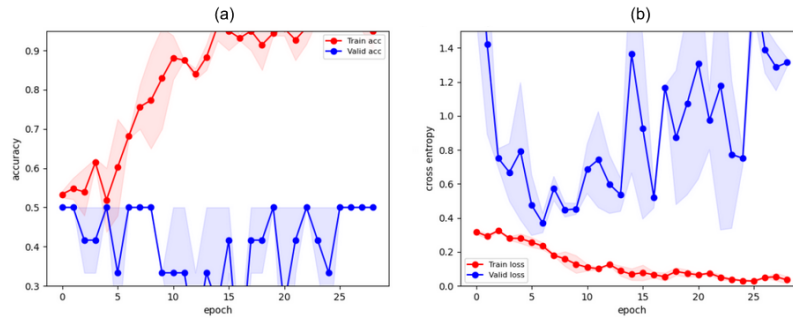


Figura 4.30: Curva de entrenamiento y validación al entrenar con el sujeto **SA012**.
Curvas de *accuracy* en (a) y curvas de *loss* en (b).

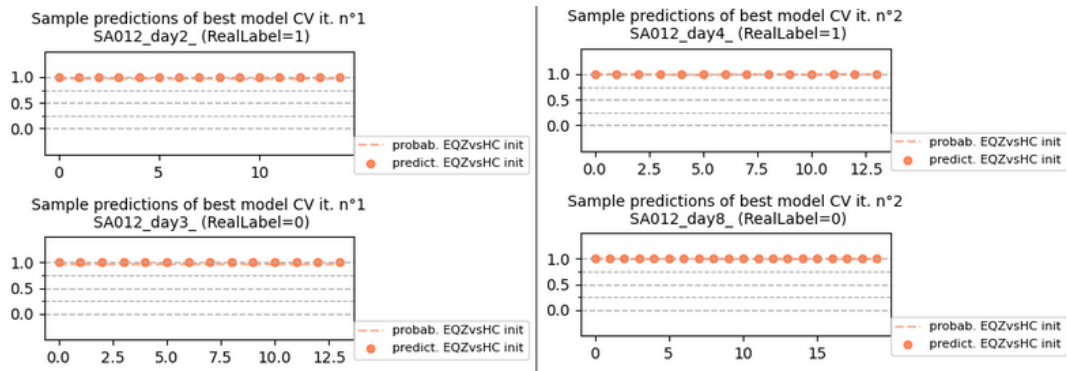


Figura 4.31: Predicciones del modelo entrenado con pesos iniciales de *EQZvsHC*, para registros EEG del sujeto **SA012**. Cada gráfico agrupa las predicciones correspondientes a un día específico. Cada punto corresponde a una predicción realizada sobre un segmento de 40 s en los registros de validación.

5. Discusión

5.1. Enfoque con extracción manual de características

El primer conjunto de características (`['IQR_Beta', 'Mean_power_Theta']`), efectivamente se muestran patrones que forman clústers, pero por otro lado, estos son muy variados según el sujeto en cuestión. Además, las variaciones en el espacio de estos supuestos clústers no siguen un desplazamiento simple entre los valores continuos de la etiqueta (i.e. los colores), lo cual los hace menos fiables.

Con respecto al segundo conjunto (`['std_Alpha', 'LZC_H-env_Complete', 'MAD_Delta', 'Mean_power_Theta']`), solo se muestran agrupaciones de puntos similares en los sujetos SA012, SA014 y SA025. Resulta particularmente relevante lo expuesto en el gráfico del sujeto SA025, puesto que en este caso si se observa un desplazamiento creciente de los colores de izquierda a derecha. Sin embargo, esto solo se presenta en ese gráfico, y a pesar de que en SA012 igualmente se forman clústers, estos a simple vista no muestran un patrón identificable.

Finalmente, al observar los *clústers* formados al analizar el tercer grupo de características (`['std_Alpha', 'IQR_Beta', 'LZC_H-pow_Complete', 'IQR_Delta', 'std_Theta']`), si se forman patrones más interesantes, en particular las gráficas de los sujetos SA007 y SA025, pues si parecen formarse continuidades en las variaciones de colores. Sin embargo, a pesar de que el resto de sujetos puede presentar muestras de clusterización, no resultan consistentes entre si.

La EQZ es una condición extensamente heterogénea, de modo que las maneras de detectar la acentuación de sus síntomas también pueden ser muy diversas. Los resultados anteriores indican que si podrían existir características comunes que entreguen información sintomatológica valiosa de los pacientes, pero que las variaciones de aquellas características podrían ser muy diferentes entre distintos sujetos. Si fuera posible identificar específicamente qué características son de interés en el rumbo sintomatológico de los pacientes, tal vez podría afinarse un método específico para cada uno.

Ahora, analizando los resultados obtenidos al usar RF con los conjuntos de características mencionados (tablas 4.1 y 4.2), se puede obtener una conclusión más objetiva acerca de la información que nos pueden aportar los conjuntos con respecto a lo que se quiere estimar, que es los niveles de sintomatología positiva en el sujeto EQZ. Por un lado, se menciona que el conjunto de características `['IQR_Beta', 'Mean_power_Theta']` no posee la suficiente información para concluir algo respecto a los síntomas positivos del paciente. Por otro lado, viendo las tablas, se observa que los otros dos conjuntos de características si nos entregan

información que puede ser valiosa para identificar los niveles sintomatológicos. Un hallazgo interesante es que a pesar de entrenar modelos para sujetos diferentes, los resultados son promisorios para casi todos los sujetos en ambos casos. Lo anterior refuerza la idea de que, a pesar de que las dinámicas de las características dentro de cada paciente puedan ser muy diferentes, si podría ser que independiente del paciente analizado, las características que deban monitorearse sean las mismas para obtener conclusiones para cualquier paciente. Por otro lado, los buenos resultados vistos al usar RF dejan en evidencia que un análisis en 2D de las características, resulta ser un análisis muy limitado sobre el verdadero potencial que presenten las características analizadas.

Los resultados anteriores deben interpretarse como preliminares, e indican que la investigación podría enfocarse en ciertas características específicas de las señales EEG, y sobretodo, cómo interactúan estas entre sí. De hecho, resulta llamativo cómo los 3 conjuntos de características seleccionados, poseen diferentes bandas de frecuencia que interactúan con información entre sí.

5.2. Enfoque con aprendizaje profundo

5.2.1. Primera aproximación con datos auxiliares

Con respecto a los *clústers* formados en el primer acercamiento con los datos *EQZvsHC* en la figura 4.11, se observa que existe una separación casi absoluta entre ambas clases. Por otro lado, viendo los gráficos que diferencian por color cada sujeto, se observa que un sujeto específico de la clase HC se infiltra en el clúster de segmentos EQZ, por lo cual, podría considerarse un sujeto *outlier*. Considerando nuevamente los gráficos que diferencian por color cada sujeto, también se observa que cada sujeto tiene su propio clúster, es decir, se hace evidente que existe una cierta heterogeneidad en los segmentos, pues estos tienden a agruparse según el sujeto desde donde son registrados. En otras palabras, se podría decir que los clústers EQZ y HC están formados por otros pequeños *subclústers*.

En la figura 4.12 se observa como la transferencia de aprendizaje acelera el proceso de ajuste del modelo, y de hecho, permite que el modelo sea más generalizable ya que el desempeño de validación alcanza un valor mayor con respecto a no usar transferencia de aprendizaje. Sin embargo, cabe hacer notar que el *accuracy* aumenta mientras el valor de pérdida también aumenta en el conjunto de validación. Para interpretar esto, se debe tener en cuenta que el *accuracy* y el *loss* no son métricas exactamente correlacionadas a la inversa. El *loss* mide la diferencia entre la probabilidad de salida del modelo, $\text{sigmoid}(f_{\theta})$, con las clases “0” y “1”, mientras que el *accuracy* mide la diferencia entre la predicción realizada (a partir de un umbral sobre la probabilidad de salida) con las clases “0” y “1”. Es decir, si varía la probabilidad de salida del modelo entonces el *loss* varía, pero el *accuracy* resulta ser más “resiliente” pues mientras la variación de la probabilidad no traspase el umbral entonces el *accuracy* no debiera cambiar. Considerando que los valores de *accuracy* tanto de los conjuntos de entrenamiento y validación aumentan, se cree que el aumento en el *loss* se debe a que el modelo a medida que mejora, a la vez está menos seguro de sus predicciones. Particularmente, esto podría deberse a que los segmentos mal clasificados del conjunto de validación (alrededor del 30%) están extremadamente mal clasificados.

Con respecto a los resultados de comparar el desempeño de los entrenamientos con o sin congelar capas, se puede ver que al congelar la primera capa del codificador la velocidad de convergencia aumenta, y es más, los valores de *loss* de validación alcanzan valores más pequeños, es decir, los segmentos muy mal clasificados dejan de estarlo. Por otro lado, dado que congelar las primeras 3 capas del codificador no entrega buenos resultados, esto quiere decir que las *features* obtenidas por la red a esa altura no necesariamente son generalizables para cualquier tarea. Dado que congelar la primera capa mejora el desempeño del modelo, esto señala que las *features* obtenidas a primer nivel del codificador si son generalizables para diferentes tareas, particularmente para ésta. Esto podría considerarse análogo al procesamiento de imágenes con redes convolucionales, donde las primeras capas se encargan de extraer características, como bordes o contornos, que resultan relevantes para cualquier tarea de identificación de imágenes.

5.2.2. Resultados con base de datos *DecompStudy*

En esta sección se analizan las curvas de entrenamiento obtenidas al entrenar el modelo con cada sujeto por separado. Recordar que se realizan 2 entrenamientos por sujeto, el primero con los pesos obtenidos en el preentrenamiento con la base de datos TUEG ([60]) como condición inicial; y el segundo con los pesos obtenidos al entrenar el modelo para discriminar entre EQZ y HC, como condición inicial.

Se observa que las curvas de *loss* en todos los casos tiende a disminuir, por lo menos en las primeras épocas de los entrenamientos, a diferencia de las curvas resultantes al entrenar para la tarea de discriminar entre EQZ y HC. Esto podría deberse a la nueva definición de *accuracy* que se hace, donde se ponderan con mayor gravedad los errores del tipo falso positivo, y por lo mismo, se modifica la ponderación de los pesos en la función de pérdida. Es un buen indicio que las curvas de *accuracy* y *loss*, en general, se vean inversamente correlacionadas. Esto es importante, pues quiere decir que la función de *loss* está siendo efectiva para lograr el objetivo, que es maximizar el *accuracy*.

En la comparación del desempeño de ambos tipos de inicialización, en lo que respecta al análisis sobre las curvas de entrenamiento de cada sujeto, se presenta una mejora tanto en el comportamiento de las curvas como en los valores máximos de *accuracy* en validación, al utilizar los pesos obtenidos preentrenando con el conjunto *EQZvsHC*.

Lo anterior indica que efectivamente es un aporte usar los pesos del modelo entrenado para discriminar entre EQZ y HC, como pesos de inicio del nuevo entrenamiento, lo cual resulta un hallazgo bastante interesante y potencialmente útil.

Al analizar los gráficos de predicción por segmento de los registros de validación (figuras 4.20-4.26), se observa que el desempeño de los modelos resultantes es bastante similar entre usar los pesos de [60] vs. los pesos de *EQZvsHC*. Es importante recordar, que un registro está bien clasificado si cumple alguna de estas 2 condiciones:

- Su etiqueta real es “1” y posee 3 o más segmentos clasificados como “1”.
- Su etiqueta real es “0” y no posee más de 2 segmentos clasificados como “1”.

Siguiendo el criterio anterior, de un total de 50 registros de validación, se tiene que los

modelos inicializados con los pesos de BENDR [60], tienen 14 registros mal clasificados, mientras que los modelos inicializados con los pesos de *EQZvsHC*, tienen también 14 registros mal clasificados. Por lo que, a pesar de lo observado en las curvas, por lo menos con los registros completos de validación, el resultado es similar entre ambas inicializaciones. En la tabla 5.1 se muestra a que sujeto pertenece cada registro mal clasificado, así como también la etiqueta verdadera (y_{real}) de cada uno.

Tabla 5.1: Detalles de los registros de validación mal clasificados según las figuras 4.20-4.26.

	BENDR init.		EQZvsHC init.	
	$y_{real} = 0$	$y_{real} = 1$	$y_{real} = 0$	$y_{real} = 1$
SA007	0	0	1	0
SA010	1	0	1	0
SA014	1	1	1	1
SA025	3	0	2	0
SA026	2	0	2	0
SA039	5	0	4	0
SA047	0	1	2	0
Total	14		14	

Con respecto a lo mostrado en la tabla, se observa que los modelos en general poseen una tendencia mayor a equivocarse con los registros con $y_{real} = 0$, es decir, con los registros obtenidos cuando el sujeto presentó síntomas positivos bajos/moderados. Esto puede deberse al hecho de que el umbral de clasificación usado es de 0.4, lo cual se traduce en que el modelo debe poseer un mayor nivel de certeza para clasificar como $\hat{y} = 0$, en comparación a la certeza requerida para clasificar como $\hat{y} = 1$. Por otro lado, cabe destacar que dentro de los registros que se clasifican mal, existen algunos casos donde esto ocurre apenas debido a solo 1 o 2 segmentos mal clasificados dentro de todo el registro, mayoritariamente al usar los pesos de *EQZvsHC*.

Por otra parte, cabe mencionar que existe variación en los desempeños de los modelos para cada sujeto, aunque esto puede deberse a un sinnúmero de razones posibles, tales como por ejemplo, un mayor ruido en la captura del registro, y quizás no pudo ser extraído completamente por la *pipeline* de limpieza previo. Además, se destaca que el sujeto SA026 posee el peor desempeño, y lo más probable es que esto sea debido al valor reducido de registros disponibles. Por el contrario, con el sujeto SA039 se obtiene un bajo rendimiento a pesar de contar con la mayor cantidad de registros. Merece la pena destacar que esta sujeto es de género femenino, pero es muy baja la muestra como para comentar cualquier suposición al respecto.

Con respecto a los últimos entrenamientos realizados, para SA011 y SA012, se observa que en el caso de SA011 el modelo queda bastante bien ajustado. La figura 4.29 muestra que solo se clasifica mal un solo registro (SA011_day5) pero es literalmente por un solo segmento mal clasificado, es decir, estuvo al borde de estar bien clasificado. Por lo tanto con esta sujeto

el modelo resulta un éxito.

Por el contrario, con el sujeto SA012, el modelo técnicamente no aprende nada, pues la curva de validación no sube del 50 %. Al observar las gráficas de predicciones por segmentos, se observa que la salida del modelo es constante. No se conoce la causa de este fenómeno, pero el resultado no es tan diferente a lo que ocurre con el SA039. Por otro lado, este caso puede deberse a la baja cantidad de registros EEG, ya que, según la tabla 5.1, los sujetos SA012 y SA026 son los que presentan un menor número de registros, lo cual podría ser la causa de su mal desempeño en ambos casos.

6. Conclusión

Se cumplieron los objetivos específicos definidos en la sección 1.4, aunque de manera parcial. Sin embargo debe tomarse en cuenta que la identificación de períodos de descompensación en sujetos padecientes de EQZ es una tarea nunca antes abordada por lo revisado en la bibliografía. Por otro lado, no existe conocimiento, por la autora de esta tesis, de la existencia de otro conjunto de datos, de un estudio longitudinal, que comprenda el registro de señales EEG acompañados de evaluaciones sintomatológicas de sujetos con EQZ.

Se desarrolló una *pipeline* de preprocesamiento para limpiar señales realmente ruidosas y con presencia de artefactos, lo cual resulta muy valioso considerando su cualidad de funcionamiento completamente automático. Esto implicaría que a futuro la tediosa tarea de preparar datos EEGs podría automatizarse y, de esta manera, reducir una considerable cantidad de tiempo. También esto podría dar pie al desarrollo de una *pipeline* de preprocesamiento en tiempo real, que permita analizar señales EEG en un horizonte de tiempo más pequeño, y por otro lado, podría incluso llegar a estandarizarse como parte de alguna metodología específica.

Con respecto al análisis de características obtenidas manualmente, se pudo descartar la existencia de correlaciones explícitas y evidentes entre pares de características exploradas (para segmentos de 40 s), al analizarlas visualmente en 2D. Estas fueron ampliamente examinadas y visualizadas de variadas formas en los gráficos y, a priori, no mostraron alguna correlación trivial. A pesar de lo anterior, si se concluye la existencia de cierta sinergia en características específicas, lo cual es evidente de manera parcial en la exploración gráfica en 2D, pero aun más evidente al utilizar modelos RF preliminares. Lo anterior, motiva a la futura exploración de la correlación potencial entre estas características y la sintomatología de los pacientes, con métodos aun más sofisticados.

Por otra parte, se desarrolló un prototipo de clasificador de intensidad de síntomas positivos en pacientes con EQZ. Si bien, su desempeño y versatilidad por sujetos no es el idóneo, si posee el potencial suficiente para ser considerado un *baseline* para futuros trabajos que aborden la misma temática. Además, en concordancia con la naturaleza heterogénea de la esquizofrenia, se mostró que el modelo si posee un desempeño prometedor para determinados sujetos.

Dentro de todo el trabajo realizado, se pudo dar cuenta de los desafíos presentes al momento de utilizar aprendizaje supervisado en las señales EEG. Uno de los problemas existentes es contar con poca cantidad de registros etiquetados, lo cual resulta aun más problemático si se consideran todas las variables adicionales asociadas, tales como la variabilidad existente al recolectar información de diferentes sujetos, las diferencias presentes en los entornos/con-

diciones en los cuales son registrados los EEG y cómo aquello repercute directamente en las señales, así como también el alto nivel de ruido presente inherentemente en las señales; entre otras variables. Este desafío se intensifica aun más al considerar una condición de naturaleza tan heterogénea como lo es la EQZ. Sin embargo, a medida que avanzan los estudios en el área, muchos aspectos podrían llegar a estandarizarse, con el objetivo de reducir la influencia de las variables recién mencionadas.

Se muestra lo provechoso que resulta emplear transferencia de aprendizaje para sobrellevar aquellas dificultades. Utilizando unos pocos registros EEG, y un número reducido de épocas, se es capaz de ajustar un modelo para que este resulte relativamente exitoso en la tarea requerida. Acompañado de esto, se reconfirma la efectiva capacidad de representación, por parte del codificador de BENDR preentrenado con el voluminoso conjunto de datos TUEG, y como esta capacidad de representación posee la suficiente generalidad para emplearse en una tarea completamente desconocida y con registros nuevos, como son los conjunto de datos *EQZvsHC* y *DecompStudy*. Además, con esto se reafirma la utilidad de estandarizar los procesos, tal como se describió en la sección 3.4.1.1.

El método de *data augmentation* propuesto en la sección 3.4.3 también logra sobrellevar, efectivamente, las limitaciones dadas por el reducido número de registros EEG disponibles para entrenar.

Un hallazgo de esta tesis es que utilizar como condiciones iniciales del modelo, los pesos de preentrenamiento de BENDR ajustados para la tarea de discriminación EQZ vs. HC, resulta más ventajoso que tan solo usar los pesos de preentrenamiento de BENDR sin ajuste como condición inicial. Esto podría dar luces de que las diferencias presentadas entre EQZ vs. HC, podrían estar correlacionadas con las diferencias presentadas en un sujeto EQZ con niveles regulados de síntomas positivos vs. el mismo sujeto con niveles incrementados de síntomas positivos; tal y como lo sugeriría la intuición.

6.1. Trabajo a futuro

Dentro de las sugerencias de investigación o continuidad dentro de la temática abordada en esta tesis, se mencionan las siguientes:

- Optimizar los costos de ejecución de la *pipeline* de preprocesamiento, con el objetivo de proyectar su potencial uso en el monitoreo de EEGs en tiempo real.
- Emplear métodos, basados en aprendizaje de máquinas, para analizar, con mayor nivel de complejidad, los conjuntos de características obtenidos en 3.3.2. Métodos sugeridos para esto son el uso de redes MLP y, especialmente, redes recurrentes (RNN). El uso de RNN podría ser de utilidad para analizar, no solo las características de cada segmento de señal, sino que cómo estas evolucionan temporalmente dentro de cada segmento. Esto ayudaría a identificar biomarcadores de la EQZ en el EEG.
- Modificar la arquitectura propuesta en 3.4.1, de tal manera que el contextualizador lineal sea rediseñado y esté compuesto por una capa de atención. Esto permitiría averiguar qué es lo que considera relevante el modelo a la hora de discriminar entre síntomas positivos regulares/bajos y síntomas positivos altos, evitando mantener el concepto de “caja negra”. Esto también, ayudaría a identificar biomarcadores de la EQZ en el EEG.

- Una vez logrado el ajuste de un modelo capaz de discriminar, de manera robusta, entre niveles de sintomatología positiva dentro de un mismo paciente; abordar la tarea de realizar predicción de estos estados. Esto, en conjunto a la identificación de biomarcadores en el EEG, correspondería a la motivación principal sobre la que se gestó esta tesis, pues resultaría ser tremendamente útil en el tratamiento de las personas padecientes de la condición de EQZ.

Bibliografía

- [1] Institute of Health Metrics and Evaluation (IHME), “Global Health Data Exchange (GHDx).” <http://ghdx.healthdata.org/gbd-results-tool?params=gbd-api-2019-permalink/27a7644e8ad28e739382d31e77589dd7>. Accessed: 09-25-2021.
- [2] Barbato, A., “Psychiatry in transition: outcomes of mental health policy shift in Italy,” *Australian and New Zealand Journal of Psychiatry*, vol. 32, no. 5, pp. 673–679, 1998.
- [3] Chernecky, C. y Berger, B., *Laboratory tests and diagnostic procedures*. Elsevier Health Sciences, 2012.
- [4] Khanna, A., Pascual-Leone, A., Michel, C., y Farzan, F., “Microstates in resting-state EEG: current status and future directions,” *Neuroscience & Biobehavioral Reviews*, vol. 49, pp. 105–113, 2015.
- [5] Michel, C. y Koenig, T., “EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: a review,” *NeuroImage*, vol. 180, pp. 577–593, 2018.
- [6] Alfimova, M. y Uvarova, L., “Changes in the EEG spectral power during perception of neutral and emotionally salient words in schizophrenic patients, their relatives and healthy individuals from the general population,” *Zhurnal vyssheĭ nervnoĭ deiatelnosti imeni I P Pavlova*, vol. 57, 2007.
- [7] Bower, B., “Synchronized thinking: Brain activity linked to schizophrenia, skillful meditation,” *Science News*, vol. 166, 2004.
- [8] Winterer, G., Ziller, M., Dorn, H., Frick, K., Mulert, C., Wuebben, Y., Herrmann, W. M., y Coppola, R., “Schizophrenia: reduced signal-to-noise ratio and impaired phase-locking during information processing,” *Clinical Neurophysiology*, vol. 111, no. 5, pp. 837–849, 2000.
- [9] Fernández, A., Méndez, M. A., Hornero, R., Ortiz, T., y López-Ibor, J. J., “Brain complexity analysis and mental disorders,” *Actas Españolas de Psiquiatría*, vol. 38, pp. 229–238, 2010.
- [10] Takahashi, T., Cho, R. Y., Mizuno, T., Kikuchi, M., Murata, T., Takahashi, K., y Wada, Y., “Antipsychotics reverse abnormal EEG complexity in drug-naïve schizophrenia: a multiscale entropy analysis,” *NeuroImage*, vol. 51, no. 1, pp. 173–182, 2010.
- [11] Wolfers, T., Doan, N. T., Kaufmann, T., Alnæs, D., Moberget, T., Agartz, I., Buitelaar, J. K., Ueland, T., Melle, I., Franke, B., Andreassen, O. A., Beckmann, C. F., Westlye, L. T., y Marquand, A. F., “Mapping the heterogeneous phenotype of schizophrenia and bipolar disorder using normative models,” *JAMA Psychiatry*, vol. 75, no. 11, pp. 1146–1155, 2018.

- [12] Torous, J., Staples, P., Barnett, I., Onnela, J.-P., y Keshavan, M., “A crossroad for validating digital tools in schizophrenia and mental health,” *NPJ Schizophrenia*, vol. 4, no. 1, pp. 1–2, 2018.
- [13] Riley, J., “Understanding metadata,” National Information Standards Organization (<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>), vol. 23, pp. 7–10, 2017.
- [14] Owen, M., Sawa, A., y Mortensen, P., “Schizophrenia,” *Lancet*, vol. 388, 2016.
- [15] Peterson, T., “Difference between positive and negative symptoms of schizophrenia.” HealthyPlace, <https://www.healthyplace.com/thought-disorders/schizophrenia-symptoms/difference-between-positive-and-negative-symptoms-of-schizophrenia>, 2018. Accessed: 01-02-2023.
- [16] Tatum, W., *Handbook of EEG interpretation*. Springer Publishing Company, 2021.
- [17] Britton, J. W., Frey, L. C., Hopp, J. L., Korb, P., Koubeissi, M. Z., Lievens, W. E., Pestana-Knight, E. M., y St. Louis, E. K., *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. American Epilepsy Society, Chicago, 2016.
- [18] Morillo, L., *Guía neurológica 7*. Asociación Colombiana de Neurología, 2005.
- [19] Sekimoto, M., Kato, M., Watanabe, T., Kajimura, N., y Takahashi, K., “Cortical regional differences of delta waves during all-night sleep in schizophrenia,” *Schizophrenia Research*, vol. 126, 2011.
- [20] Kelsom, C. y Lu, W., “Development and specification of gabaergic cortical interneurons,” *Cell & bioscience*, vol. 3, 2013.
- [21] Cortes-Briones, J. A., Tapia-Rivas, N. I., D’Souza, D. C., y Estevez, P. A., “Going deep into schizophrenia with artificial intelligence,” *Schizophrenia Research*, vol. 245, pp. 122–140, 2022. *Computational Approaches to Understanding Psychosis*.
- [22] Alzahy, A., Elgammal, M., Mohammed, H., y Mostafa, H., “Optimal eeg window size for neural seizure detection,” en *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, pp. 1–4, IEEE, 2019.
- [23] Kanoga, S. y Mitsukura, Y., “Review of artifact rejection methods for electroencephalographic systems,” en *Electroencephalography* (Sittiprapaporn, P., ed.), cap. 6, Rijeka: IntechOpen, 2017.
- [24] Jiang, X., Bian, G.-B., y Tian, Z., “Removal of artifacts from eeg signals: a review,” *Sensors*, vol. 19, no. 5, p. 987, 2019.
- [25] Lempel, A. y Ziv, J., “On the complexity of finite sequences,” *IEEE Transactions on Information Theory*, vol. 22, 1976.
- [26] Ziv, J. y Lempel, A., “A universal algorithm for sequential data compression,” *IEEE Transactions on information theory*, vol. 23, no. 3, pp. 337–343, 1977.
- [27] Naqa, I. E. y Murphy, M., *Machine Learning in Radiation Oncology*. Springer, 2015.
- [28] Mitchell, T., *Machine learning*. McGraw-hill New York, 1997.
- [29] Gurney, K., *An introduction to neural networks* (1st ed.). CRC press, 1997.
- [30] Tapia, N., “Un modelo secuencial profundo para detectar eventos en el electroencefalogramograma,” 2023.

- grama del sueño,” Tesis de Magíster de Ingeniería Civil Eléctrica, Universidad de Chile, 2022.
- [31] Bishop, C., Pattern recognition and machine learning.
- [32] Cybenko, G., “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [33] Nielsen, M., “A visual proof that neural nets can compute any function.” <http://neuralnetworksanddeeplearning.com/chap4.html>, 2016. Accessed: 12-25-2022.
- [34] Leshno, M., Lin, V. Y., Pinkus, A., y Schocken, S., “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural networks*, vol. 6, no. 6, pp. 861–867, 1993.
- [35] Sra, S., Nowozin, S., y Wright, S., Optimization for machine learning. Mit Press, 2012.
- [36] Bottou, L., “Online algorithms and stochastic approximations,” en *Online Learning and Neural Networks* (Saad, D., ed.), Cambridge, UK: Cambridge University Press, 1998.
- [37] LeCun, Y., Bengio, Y., y Hinton, G., “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [38] Takahashi, D., “Nvidia ceo bets big on deep learning and vr,” *VentureBeat*. Apr, vol. 5, 2016.
- [39] Thompson, N., Greenewald, K., Lee, K., y Manso, G., “The computational limits of deep learning,” MIT Initiative on the digital economy research brief 2020 Vol. 4, 2020.
- [40] Hendrycks, D. y Gimpel, K., “Gaussian error linear units (gelus),” 2020.
- [41] Miyakoshi, M., “Makoto’s preprocessing pipeline. (n.d.)” https://scn.ucsd.edu/wiki/Makoto's_preprocessing_pipeline. Accessed: 07-07-2022.
- [42] Ronneberger, O., Fischer, P., y Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” en *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [43] Kothe, C., Miyakoshi, M., y Delorme, A., “clean_rawdata.” https://github.com/scn/clean_rawdata, note = Accessed: 07-07-2022, 2019.
- [44] Mullen, T., Kothe, C., Chi, M., Ojeda, A., Kerth, T., Makeig, S., Jung, T.-P., y Cauwenberghs, G., “Real-time neuroimaging and cognitive monitoring using wearable dry eeg,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2553–2567, 2015.
- [45] Perrin, F., Pernier, J., Bertrand, O., y Echallier, J. F., “Spherical splines for scalp potential and current density mapping,” *Electroencephalography and clinical neurophysiology*, vol. 72, no. 2, pp. 184–187, 1989.
- [46] Hunsley, J. y Mash, E., A guide to assessments that work. Oxford University Press, 2008.
- [47] Opler, M., Yavorsky, C., y Daniel, D., “Positive and negative syndrome scale (panss) training: challenges, solutions, and future directions,” *Innovations in clinical neuroscience*, vol. 14, no. 11-12, p. 77, 2017.
- [48] Andreasen, N., “Scale for the assessment of positive symptoms (saps),” University of Iowa, 1984.
- [49] Norman, R., Malla, A., Cortese, L., y Diaz, F., “A study of the interrelationship bet-

- ween and comparative interrater reliability of the saps, sans and panss,” *Schizophrenia Research*, vol. 19, no. 1, pp. 73–85, 1996.
- [50] Lyne, J., Kinsella, A., y O’Donoghue, B., “Can we combine symptom scales for collaborative research projects?,” *Journal of psychiatric Research*, vol. 46, no. 2, pp. 233–238, 2011.
- [51] Erp, T. V., Preda, A., Nguyen, D., Faziola, L., Turner, J., Bustillo, J., Belger, A., Lim, K., McEwen, S., Voyvodic, J., Mathalon, D., Ford, J., Potkin, S., y Fbirn, “Converting positive and negative symptom scores between panss and saps/sans,” *Schizophrenia research*, vol. 152, no. 1, pp. 289–294, 2014.
- [52] Olejarczyk, E. y Jernajczyk, W., “Graph-based analysis of brain connectivity in schizophrenia,” *PloS one*, vol. 12, no. 11, p. e0188629, 2017.
- [53] Oh, S. L., Vicnesh, J., Ciaccio, E., Yuvaraj, R., y Acharya, R., “Deep convolutional neural network model for automated diagnosis of schizophrenia using eeg signals,” *Applied Sciences*, vol. 9, no. 14, 2019.
- [54] Krishnan, P. T., Raj, A. N. J., y Balasubramanian, P., “Schizophrenia detection using multivariate empirical mode decomposition and entropy measures from multichannel eeg signal,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 1124–1139, 2020.
- [55] Singh, K., Singh, S., y Malhotra, J., “Spectral features based convolutional neural network for accurate and prompt identification of schizophrenic patients,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 235, no. 2, pp. 167–184, 2021.
- [56] Olejarczyk, Elzbieta, Jernajczyk, y Wojciech, *EEG in schizophrenia*. RepOD, 2017, [10.18150/repod.0107441](https://doi.org/10.18150/repod.0107441).
- [57] Zwillinger, D. y Kokoska, S., *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.
- [58] Urrutia, J., Estévez, P., y Vergara, J., “Subset feature selection with structural variables,” en *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–6, 2021.
- [59] McInnes, L., Healy, J., y Melville, J., “Umap: Uniform manifold approximation and projection for dimension reduction,” *Journal of Open Source Software*, 2018.
- [60] Kostas, D., Aroca-Ouellette, S., y Rudzicz, F., “Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data,” *Frontiers in Human Neuroscience*, vol. 15, p. 653659, 2021.
- [61] SPOClab-ca, “Bendr.” <https://github.com/SPOClab-ca/BENDR>, note = Accessed: 12-02-2023, 2021.
- [62] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., y Polosukhin, I., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [63] Wu, Y. y He, K., “Group normalization,” en *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- [64] Kostas, D. y Rudzicz, F., “Dn3: An open-source python library for large-scale raw neurophysiology data assimilation for more flexible and standardized deep learning,” *bioRxiv*, 2021.

xiv, pp. 2020–12, 2020.

- [65] Obeid, I. y Picone, J., “The temple university hospital eeg data corpus,” *Frontiers in neuroscience*, vol. 10, p. 196, 2016.
- [66] MATLAB, version 9.12.0 (R2022a). Natick, Massachusetts: The MathWorks Inc., 2022.
- [67] Delorme, A. y Makeig, S., “Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis,” *Journal of neuroscience methods*, 2004.
- [68] Rossum, G. V. y Drake, F., *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2022.
- [69] Aboy, M., Hornero, R., Abásolo, D., Álvarez, D., y Daniel, “Interpretation of the lempel-ziv complexity measure in the context of biomedical signal analysis,” *IEEE transactions on biomedical engineering*, vol. 53, no. 11, pp. 2282–2288, 2006.
- [70] Hornero, R., Aboy, M., Abásolo, D., McNames, J., y Goldstein, B., “Interpretation of approximate entropy: analysis of intracranial pressure approximate entropy during acute intracranial hypertension,” *IEEE transactions on biomedical engineering*, vol. 52, no. 10, pp. 1671–1680, 2005.

Anexos

Anexo A. Ilustración criterio de etiquetado por sujeto

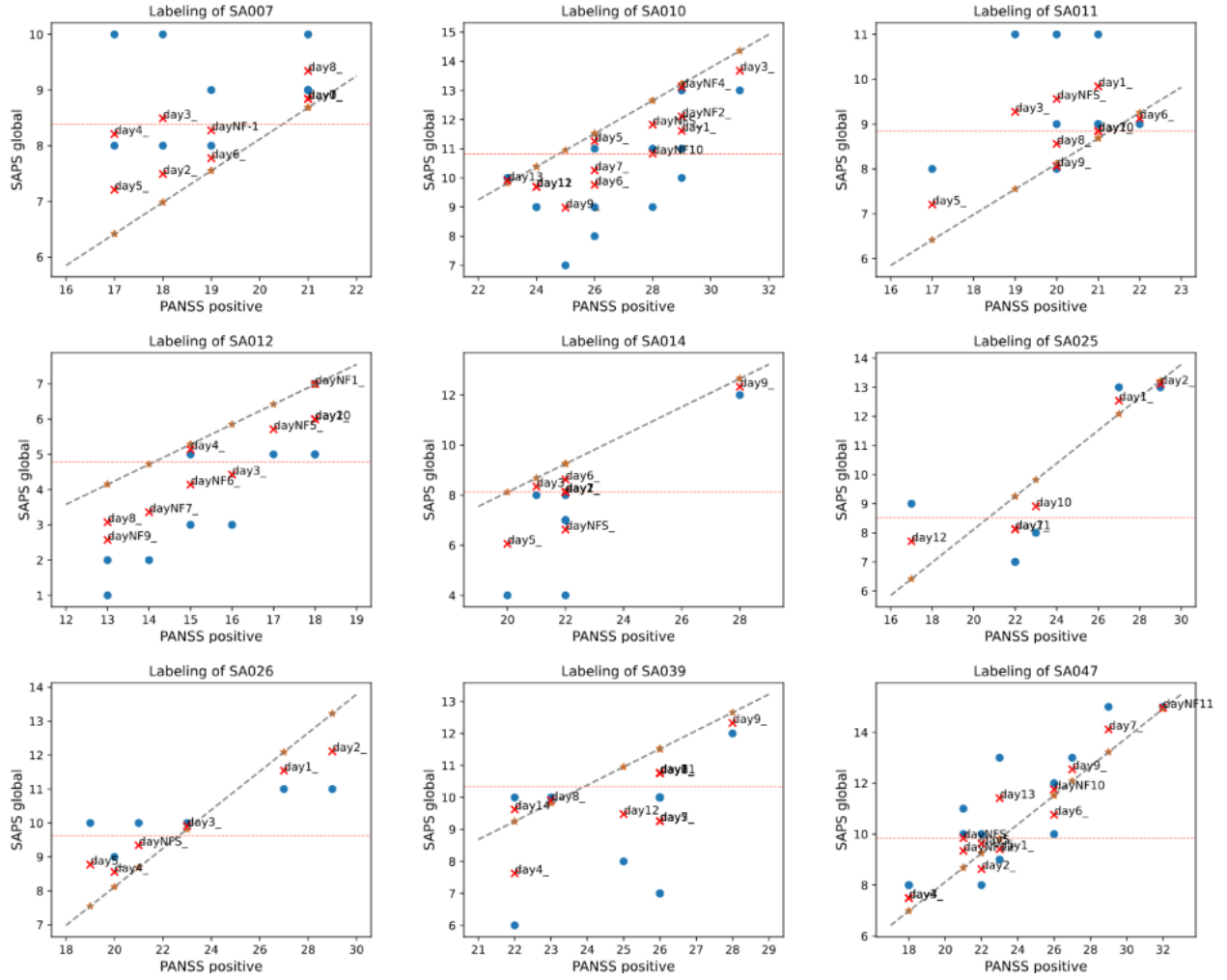


Figura A.1: Ilustración del proceso utilizado para etiquetar las señales EEG para cada sujeto; en cada recuadro está representada la recta de la ec. (3.2) por una línea punteada gris. Los puntos azules corresponden a los valores explícitos $SAPS_{global}$ y $PANSS_{positive}$, según los metadatos registrados. Las estrellas anaranjadas representan el valor del \widehat{SAPS}_{global} estimado dado el valor $PANSS_{positive}$ registrado en los metadatos. Luego, las cruces rojas representan el valor $SAPS_{def}$, es decir, el valor promedio entre $SAPS_{global}$ y \widehat{SAPS}_{global} . Finalmente, la línea punteada roja representa el umbral binarizador, dado por la mediana entre los valores \widehat{SAPS}_{global} de cada sujeto.

Anexo B. Sanity checks LZC

En esta sección se describen los experimentos realizados para comprobar el buen funcionamiento del algoritmo que calcula LZC implementado para esta tesis. Por buen funcionamiento se refiere a que los resultados entregados tengan concordancia con lo que debiera representar la métrica según su definición en la teoría, lo cual corresponde al nivel de aleatoriedad o complejidad de una señal. El código de la implementación de estos experimentos, y cálculo de LZC, se encuentra disponible en <https://github.com/frannfuri/lempelziv-testing>.

Se trabaja con señales simuladas, a las cuales se les computa el valor de su LZC, utilizando como umbral la mediana de cada señal⁸. Estos experimentos son realizados para observar la variación de la métrica de complejidad frente a ciertos escenarios, y de esta manera, comprobar que su interpretabilidad tenga relación con el nivel de complejidad de cada señal, que es lo que se debería estar midiendo. Dado que las señales fisiológicas, como el EEG, corresponden a señales análogas que son digitalizadas, es que se debe asumir que las siguientes señales sintéticas son representaciones digitales de señales análogas; es decir, cada valor de señal $x(t)$, $t = 1, \dots, T$ se asume que representa a una señal fisiológica $x(\tau)$, en el instante del tiempo $\tau_t = t \frac{1}{f_s}$, donde f_s corresponde a la frecuencia de muestreo correspondiente.

El primer experimento consiste en generar cinco señales sintéticas diferentes. La primera corresponde a una sinusoidal simple; a esta misma señal se le aplica una modulación de amplitud, lo cual resulta en la segunda señal. Luego, a la misma primera señal se le aplica una modulación de frecuencia, lo cual corresponde a la tercera señal. Por otro lado, la cuarta señal corresponde a la misma señal sinusoidal del principio, pero esta vez se le adiciona ruido blanco. Finalmente, la quinta señal corresponde simplemente a ruido blanco, sin ninguna componente sinusoidal. Estas cinco señales poseen una duración de 5 s, con $f_s = 125 \text{ Hz}$, y a cada señal completa se le computa su LZC. Las ilustraciones de estas señales acompañadas de su valor LZC resultante se pueden observar en la figura B.1.

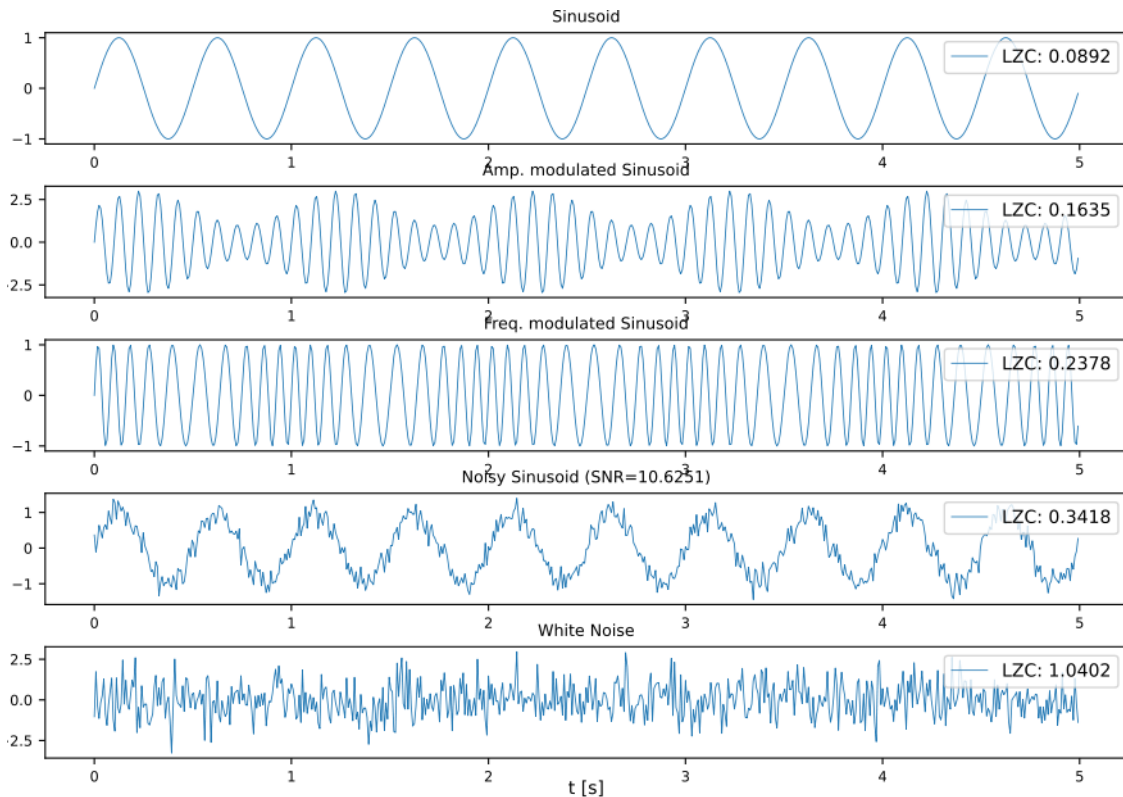


Figura B.1: Señales generadas y su respectivo valor calculado de *LZC-median*.

En la figura B.1 se observa que la primera señal, la cual representa una señal periódica y la más simple de todas, presenta el nivel más bajo de LZC con respecto a las otras. Al

⁸ Esto, en la sección 3.3.1, equivaldría a la métrica *LZC-median*.

adicionar algún tipo de modulación se observa que el valor de complejidad resultante aumenta. Sin embargo, debido a que la modulación agregada en ambos casos es debido a un componente periódico, resulta que el aumento no es tan notorio, lo cual es correcto. Por otro lado, al agregar un componente de ruido a la señal sinusoidal, se eleva aun más la LZC, como sería de esperar, pues el componente adicionado no presenta correlación entre sus puntos. Finalmente, se muestra que una señal sin ningún componente periódico y sin ningún tipo de correlación estadística, presenta el valor más alto de LZC en comparación a los otros casos, lo que confirma que en este experimento la métrica si captura el nivel de aleatoriedad de la señal, que es lo que se desea.

El segundo experimento, toma como referencia algunas de las experiencias realizadas en el paper “*Interpretation of the Lempel-Ziv Complexity Measure in the Context of Biomedical Signal Analysis*” [69], las cuales se pretende replicar, para así confirmar la implementación de la LZC en esta tesis. Estas experiencias consisten en analizar la LZC de señales sintéticas simuladas, que ya han sido usadas en otra ocasión para estudiar la métrica de *approximate entropy*, que cuantifica la irregularidad [70]. En este caso se analiza no solo la LZC de cada señal, sino que cómo esta evoluciona al realizar múltiples mediciones en segmentos de la señal (i.e., ventanas deslizantes).

A continuación se describen la seis señales sintéticas que se simularon, acompañadas de su ilustración en la figura B.2:

1. **LZC vs frecuencia:** Se genera una señal de amplitud constante pero de tipo chirrido, la cual va aumentando su frecuencia de manera lineal desde 0.5 Hz a 5 Hz en un intervalo de 40 s .
2. **LZC vs amplitud:** Se genera la misma señal anterior tipo chirrido, pero además se modula su amplitud con una componente sinusoidal.
3. **LZC vs número de componentes de frecuencia:** Se generaron 4 señales periódicas de 10 s de duración con 1, 2, 5 y 7 componentes de frecuencia respectivamente. Las 4 señales se concatenaron.
4. **LZC vs tasa-síñal-a-ruido (SNR):** Se genera una señal sinusoidal de 40 s a la cual se le adiciona ruido. La idea es que cada 10 s la SNR aumenta su valor.
5. **LZC vs potencia del ruido:** Se genera una secuencia de 40 s de ruido blanco Gaussiano con *steps* de aumento de potencia cada 10 s (0.1, 0.3, 0.5 y 0.7).
6. **LZC vs ancho de banda del ruido:** Se genera una señal de 40 s compuesta por segmentos de ruido coloreado que van aumentando su ancho de banda espectral.

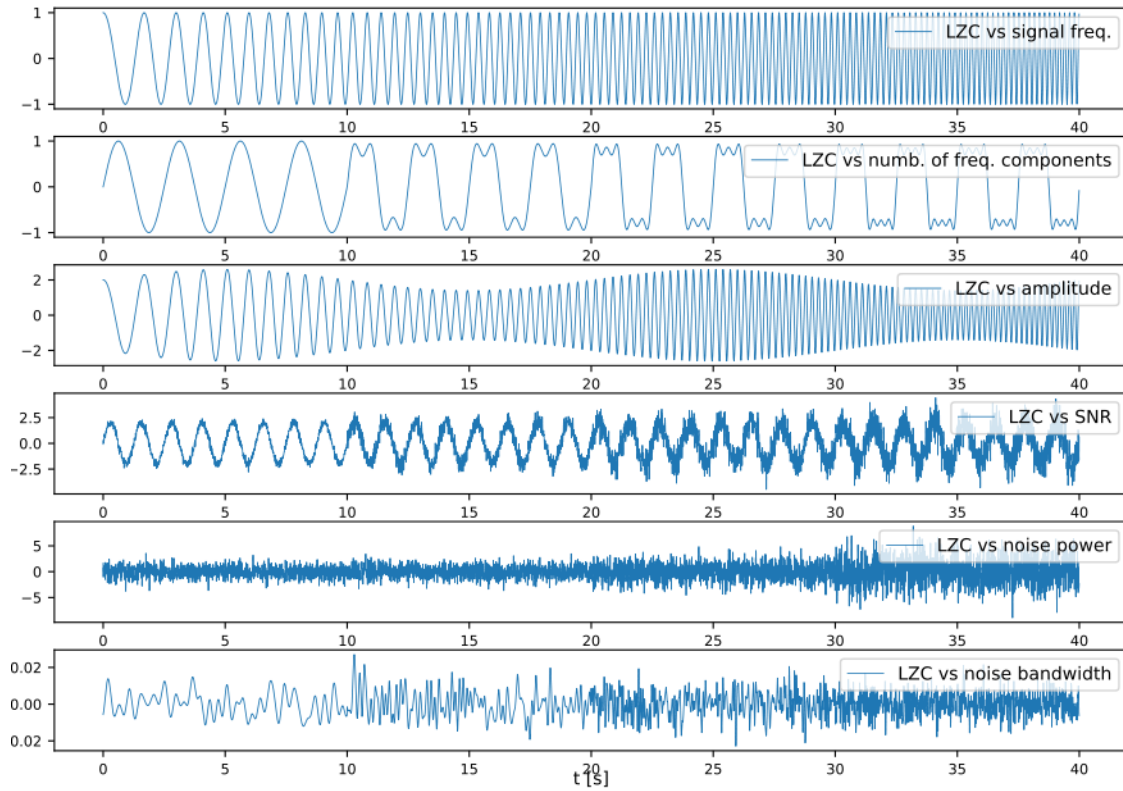


Figura B.2: Señales generadas a las cuales se les computa la LZC por ventana deslizante de 10 s.

En los seis casos, para cada señal, se calcula la LZC en ventanas consecutivas de 10 s con traslape de 90 %. Los resultados de estas mediciones se presentan en la figura B.3, los cuales son graficados para facilitar la observación de su dinámica en el tiempo.

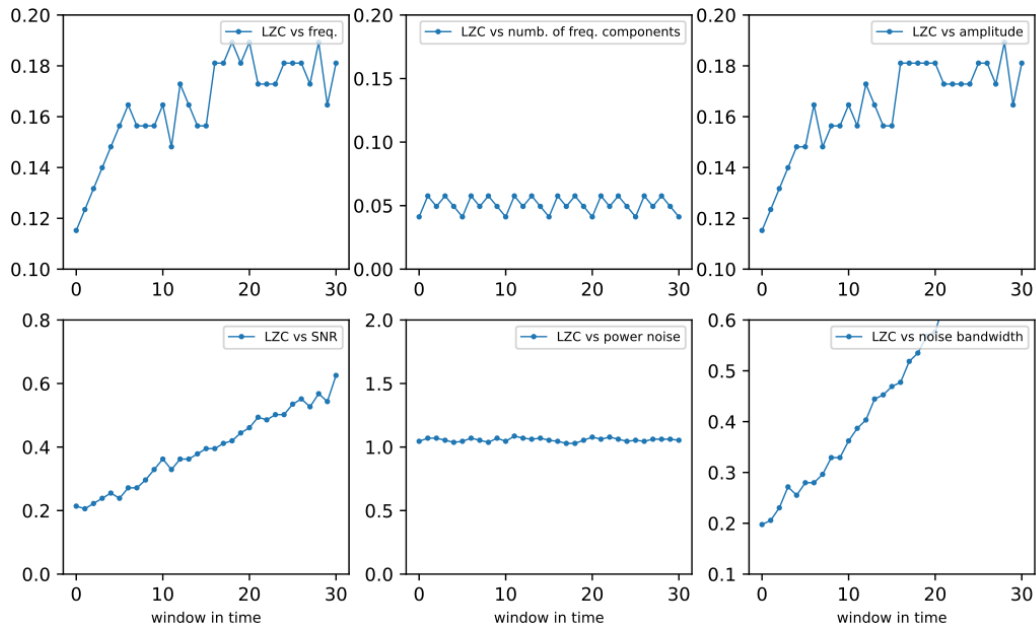


Figura B.3: Evolución del valor de LZC por ventana deslizante (10 s con traslape del 90 % entre sí) a lo largo de las señales descritas en B.2.

En la figura B.3, en “*LZC vs freq*” se observa que la LZC se incrementa al aumentar la frecuencia de una senoide. Por el contrario, en “*LZC vs amplitude*” se ve que la modulación por amplitud resulta no generar incremento en la LZC. Cabe destacar que, además, se reafirma que la señal tipo chirrido presenta valores de LZC considerablemente mayores con respecto a la señal sinusoidal de frecuencia constante que se presenta en la figura B.1; lo cual coincide con lo esperado.

Siguiendo con la figura B.2, se observa en “*LZC vs numb. of freq. components*”, que la LZC no es sensible al número de armónicos presentes en la señal periódica, lo cual concuerda con lo observado en [69].

Por otro lado, con respecto al ruido se observa un aumento en la LZC al aumentar la relación SNR como se ve en “*LZC vs SNR*”, lo cual es lo esperado. Además, por lo visto en “*LZC vs noise bandwidth*”, se observa que la LZC es sensible al ancho de banda de la señal. Sin embargo, se encuentra la sorpresa de que la LZC no se ve afectada por la potencia del ruido, como se ve en “*LZC vs power noise*”, sin embargo, esto coincide con lo mostrado en el paper [69], de donde deducen que existe una dependencia entre la LZC y la potencia del ruido a un alto nivel de SNR, pero que la LZC incrementaría solo hasta un cierto punto de saturación.

Finalmente, todo lo mostrado en la figura B.3 concuerda de manera importante con los resultados mostrados en [69], lo cual reafirma que la métrica LZC implementada para esta tesis si está capturando lo que corresponde de las señales.