Análisis de Señales EEG Mediante Algoritmos Clasificadores Usando el Hardware OpenBCI

Presentado por:

Lina Marcela Chavarro Losada

Andres Eduardo Rivera Gomez

Trabajo escrito de grado

Corporación Universitaria del Huila - Corhuila

Programa de Ingeniería Mecatrónica

Neiva, Huila

2023

Análisis de Señales EEG Mediante Algoritmos Clasificadores Usando el Hardware OpenBCI

Lina Marcela Chavarro Losada

Andres Eduardo Rivera Gomez

Trabajo escrito de grado

director:

Jorge Luis Aroca Trujillo

Corporación Universitaria del Huila - Corhuila

Programa de Ingeniería Mecatrónica

Neiva, Huila

2023

Resumen

La revolucionaria interfaz cerebro-ordenador (BCI) ha transformado la comunicación cerebral al capturar señales electroencefalográficas (EEG) para su posterior análisis, abriendo así un vasto espectro de novedosas aplicaciones en la interacción cerebral con el entorno. En este contexto innovador, se utiliza la placa Cyton Biosensing Board (8-channels) desarrollada por OpenBCI, que proporciona los datos necesarios para representar y descifrar comandos de control a partir de señales cerebrales. Para lograr esto, se implementa un proceso que incluye filtración de señales entre 1-30 Hz, extracción de características en el dominio tiempo y frecuencia y un algoritmo clasificador basado en SVM (Máquinas de Soporte Vectorial). Los resultados obtenidos y las métricas evacuativas presentadas en este trabajo pueden influir en el desarrollo de futuras aplicaciones tecnológicas, que puedan ser usadas para comandos de sillas de ruedas y brazos robóticas controlados por la mente.

Índice

Índice de figuras

Índice de tablas

Sección 1

1. INTRODUCCION

1.1 Motivación

La inteligencia artificial es el resultado de la necesidad en la búsqueda de toma de decisiones de una forma autónoma y eficaz, similares a la obra humana. Dentro de los diversos campos que abarca la inteligencia artificial, se maneja un término conocido como machine Learning. Los primeros algoritmos de machine learning aparecieron en 1950. El machine learning o aprendizaje automático es a la vez una tecnología y una ciencia («data science») que permite que un ordenador realice un proceso de aprendizaje sin haber sido previamente programado para ello. Esto es posible gracias a que el ordenador se vuelve capaz de identificar "patterns" (patrones de repeticiones estadísticas) y extraer las predicciones estadísticas. Lo dicho anteriormente, se realiza mediante una exploración de una cantidad considerable de datos, lo cual es factible para numerosas aplicaciones que involucren en análisis de datos. Tal es el caso de aplicaciones de reconocimiento de patrones señales EEG para el uso de la interfaz cerebro-máquina (Brain Machine Interface o BMI). Los sistemas BMI consisten en un protocolo de comunicación entre el cerebro de una persona humana con una máquina, extrayendo patrones de onda generados por la actividad del cerebro, con el objetivo de enviar una orden al robot y en este caso particular, desarrollar comandos que pueden ser usados para el movimiento de un vehículo. Esta aplicación puede tener un enfoque que mejore la calidad de vida de personas con limitaciones motoras severas, el cual sufren de deterioros en el sistema nervioso y muscular, pero no presentan lesiones cerebrales, permitiendo la aplicación de un sistema BMI mediante una prótesis que responde a las señales eléctricas originadas en el lóbulo frontal.

He allí que en el presente documento se centra de modo puntual en la recolección y procesamiento de datos para su posterior implementación de comandos mediante el uso de machine Learning.

1.2 Descripción del problema

Debido a la escalabilidad que puede tener el desarrollo de sistemas de comandos mentales, se puede abordar desde diferentes aplicaciones que mejoren la calidad de vida de las personas. Estas aplicaciones pueden llegar a ser un sistema genérico y escalable que permita la captura y procesamiento de señales cerebrales para generar comandos en tiempo real que van a ser usados en vehículos terrestres, aéreos, brazos robóticos, entre otros. Sin embargo, como aporte a la sociedad, se puede considerar soluciones a la falta de accesibilidad y movilidad que enfrentan las personas con limitaciones físicas que están en sillas de ruedas. En muchos casos las personas con limitaciones buscan ayuda para acceder a edificios, calles y espacios públicos, lo que significa que estas personas pueden tener dificultades de independencia para desplazarse y participar en actividades cotidianas.

1.3 Objetivos

Objetivo general

Implementar un análisis de señales electroencefalográficas basado en OpenBCI y algoritmos de clasificación de señales.

Objetivos específicos

- Generar la base de datos de las señales electroencefalográficas mediante el hardware OpenBCI.

- Preprocesar las señales electroencefalográficas para que los datos sean propicios para la clasificación usando la reducción de dimensionalidad por el método de análisis de componentes principales y conceptos de covarianza y correlación de las características halladas.
- Realizar un algoritmo multicategorico para clasificar los cinco comandos a usar, mediante el método de máquina de vector de soporte.
- Evaluar el modelo con métricas, matrices de confusión y gráficas ROC.

1.4 Contribuciones

Considerando los objetivos establecidos y el contexto del proyecto en el que se desarrolla, las contribuciones realizadas se presentan a continuación:

- Algoritmo clasificador de señales EEG multiclase mediante el hardware OpenBCI
- Cuadros comparativos de evaluación del algoritmo

1.5 Organización del documento

El documento esta organizado de la siguiente forma:

- En la sección 2, se encuentra el estado del arte en cuanto a proyectos que han realizado recolección de señales electroencefalográficas durante la acción de movimientos e imaginación motora. Se exponen de forma breve los filtros de señal más usados en proyectos de investigación, al igual del concepto y descripción del proceso de extracción de características, explicando las más usadas. Posteriormente se exponen las herramientas para el análisis de las características y finalmente se describe el proceso de clasificación que se proyectó usar.

- En la sección 3, se presenta la metodología orientada a dar solución a cada uno de los objetivos planteados.
- En la sección 4, se muestran todos los resultados respectivos a cada planteamiento metodológico mediante métricas de evaluación, donde se exponen las gráficas y tablas obtenidas.
- En la sección 5, se muestran las conclusiones obtenidas a partir de los resultados.
- En la sección 6, se proponen trabajos futuros para contribuir al desarrollo y validación del modelo del proyecto

Sección 2

2. MARCO TEÓRICO

Estimar los estados cognitivos o afectivos a partir de señales cerebrales es un paso clave pero desafiante en la creación de aplicaciones de interfaz pasiva cerebro-computadora (BCI)(Appriou et al., 2020)

Los algoritmos de aprendizaje automático tienen el potencial de automatizar el análisis de EEG(Soangra et al., 2023). Los BCI funcionan detectando patrones en la actividad cerebral que corresponden a movimientos o comandos específicos. Por ejemplo, si una persona es inestable al caminar, ciertas áreas de su cerebro se activarán, generando patrones específicos de actividad eléctrica que pueden ser detectados por un BCI. (Soangra et al., 2023)

Por esta razón, mediante la creación de algoritmos de aprendizaje para un conjunto de datos recolectados a partir del estudio de EEG, resulta ser el proceso de estudio que consta de la investigación de una adecuada obtención de datos EEG y su debido preprocesamiento para su posterior análisis y clasificación.

2.1 Recolección de datos

La electroencefalografía (EEG) es el método más común para extraer información relevante de la actividad cerebral debido a su alta resolución temporal, bajo costo, portabilidad y bajo riesgo para el usuario. Sin embargo, los electrodos del cuero cabelludo tienen graves desventajas, como la no estacionariedad, la baja relación señal-ruido y la mala resolución espacial. Además, se requieren habilidades de usuario adecuadas para implementar protocolos clínicos, de modo que el BCI debe participar en condiciones de laboratorio controladas como los son las imágenes motoras (MI). Como resultado, los procedimientos de adquisición de señales, instrumentación y desarrollo de software deben integrarse de

manera efectiva para validar los protocolos clínicos para las respuestas neuronales del cerebro.(Cardona-Álvarez et al., 2023)

A continuación, se exploran diversos métodos de recolección de señales electroencefalográficas utilizados en antiguos proyectos de investigación.

Alazrai y sus colegas realizaron un respectivo procedimiento para la recolección de datos, mediante dos grupos de sujetos, donde el grupo S corresponde a 18 individuos sanos (seis mujeres y doce hombres, con una desviación estándar de edad media de 21.2 a 2.9 años, cuatro zurdos y catorce diestros). (Alazrai et al., 2019)

Por otro lado, el grupo A es aquel que consta de cuatro personas masculinas, quienes presentan amputaciones transradiales. Este grupo se expone con una desviación estándar de edad media de 28.5 - 6.2 años, anexo a esto, se ha recolectado información relevante de los sujetos amputados por vía transradial con referencia a datos claves como la mano dominante, la mano amputada, tiempo transcurrido desde la amputación y el porcentaje que equivale al antebrazo restante.(Alazrai et al., 2019)

Por consiguiente, el procedimiento experimental consistió en que cada sujeto se sentara en una silla y relajará los brazos sobre una mesa situada frente a él. En la mesa se colocó una pantalla de ordenador a una distancia de aproximadamente 50 cm del sujeto y se empleó para mostrar diversas señales visuales. En concreto, cada pista visual notifica al sujeto que imagine la realización de una tarea manual específica. Las tareas de IM consideradas pueden agruparse en cuatro categorías: (1) Reposo (M1), (2) Tareas relacionadas con el agarre, que incluye el agarre de diámetro pequeño (M2), el agarre lateral (M3) y el agarre de tipo extensión (M4), (3) Tareas relacionadas con la muñeca, que incluyen la desviación cubital y radial (U/R) de la muñeca (M5) y la flexión y extensión (F/E) de la muñeca (M6), y (4) Tareas relacionadas con los dedos, incluyendo la F/E del dedo índice (M7), la

Subjeto	Dominante	Mano amputada	Tiempo transcurrido desde la amputación (años)	Porcentaje restante del antebrazo
A_1	RH	LH	3.5	28 %
A_2	RH	RH	1.5	10 %
A_3	LH	LH	4	95 %
A_4	RH	RH	5	37 %

Cuadro 1

Información concreta recolectada de parte de los sujetos con amputación vía transradial.

Fuente: Alazrai et al., 2019

F/E del dedo corazón (M8), la F/E del dedo anular (M9), la F/E del dedo meñique (M10) y la F/E del dedo pulgar (M11). La figura 4 muestra imágenes de ejemplo de las señales visuales asociadas a cada tarea de imagen de la mano.(Alazrai et al., 2019)

En la pantalla del ordenador es mostrada una señal visual por tres segundos. Luego se muestra una pantalla negra para pedir al sujeto que empiece a imaginar que realiza la tarea asociada a la señal visual mostrada. Al grupo de las personas sanas, se les pidió que realizaran la tarea con la mano derecha, mientras que a las personas con amputación vía transradial, la emplean con la mano izquierda, todo este proceso teniendo los ojos cerrados. El número de ensayos registrados por cada tarea de mano imaginaria para cada sujeto sano es de 40 ensayos, mientras que el número de ensayos registrados por cada tarea de mano imaginaria para cada sujeto con amputación transradial es de 56 ensayos. En particular, la duración de la señal visual, registrada durante el descanso, las tareas relacionadas con la

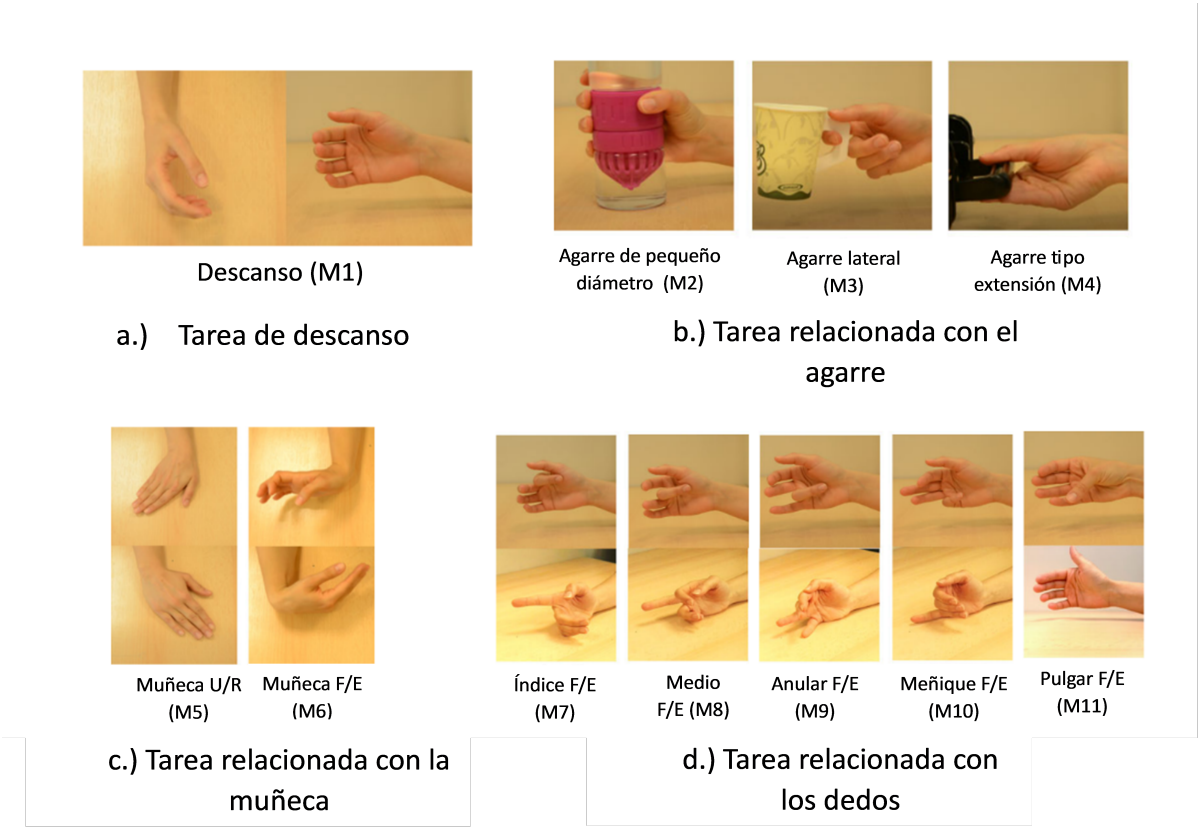

Figura 1

Imagenes de muestra de las señales visuales empleadas por cada tarea de IM.

Fuente: Alazrai et al., 2019

muñeca y las tareas relacionadas con los dedos es igual a 10s. Por otro lado, la duración de cada ensayo, incluida la duración de la señal visual, asociada a las tareas de agarre de diámetro pequeño, agarre lateral y agarre de tipo extensión es igual a 14 s, 14 s y 12 s, respectivamente.(Alazrai et al., 2019)

Granata y sus colegas incluyen los resultados de cinco amputados transradiales izquierdos consecutivos. Se emplearon registros electromiográficos de superficie para dirigir los

comandos motores a la prótesis. Durante el período de entrenamiento, cada paciente realizó diferentes tareas con el objetivo de reconocer las propiedades físicas de los objetos (incluyendo forma, consistencia y textura), realizando una variedad de movimientos simples y complejos y produciendo diferentes niveles de fuerza y presión. Además, muchas horas de estimulación nerviosa se dedicaron al mapeo topográfico psicofísico de la entrada sensorial y a validar su estabilidad en el tiempo.(Granata et al., 2020)

Los pacientes se sometieron a un examen neurofisiológico y de neuroimagen multimodal, incluida la electroencefalografía (EEG), el registro de las respuestas del EEG a la estimulación magnética transcraneal de la corteza motora (TMS-EEG) y la resonancia magnética estructural y funcional (MRI y fMRI).(Granata et al., 2020)

Ali y sus colegas utilizaron el auricular EMOTIV™ Insight de 5 canales con sensores de polímero semi seco tiene una frecuencia de muestreo interna de 128 muestras por segundo por canal que se utilizó para los experimentos. Esta herramienta fue construida especialmente para permitir el uso de BCI y la investigación, filtrando las señales y enviando los datos de forma inalámbrica a la computadora, lo que ofrece portabilidad. Este auricular EEG móvil ofrece detección de todo el cerebro y electrónica avanzada para producir señales limpias y fuertes.(Ali et al., 2021)

Para la recolección de datos, se observó un conjunto de datos dentro de los cuales se encuentran 9 sujetos: cinco discapacitados y cuatro sujetos sanos. Se les permitió observar seis objetos diferentes a través de la pantalla del ordenador. Las imágenes se mostraron en orden arbitrario. Cada imagen duró 0.1 segundos, seguidos de 0.3 segundos de intervalo. En

total se realizaron 4 sesiones y cada sesión consta de seis tareas; cada tarea equivale a cada objeto (Im- Age). En cada una de las tareas se aplicó la siguiente regla: 1. Anotar el número de imágenes que parpadean. 2. Desenlace de las seis imágenes con un tono de alarma. 3. Se excitó un orden aleatorio de parpadeos tras 4 segundos de tono de alarma, seguido del registro de la señal EEG. Se seleccionó un número aleatorio entre 20-25 como bloque. 4. Los datos se dedujeron mediante una técnica de aprendizaje automático. 5. Al final de cada tarea/ejecución, se pidió a los sujetos la secuencia contada. La duración de la tarea era de unos 60 segundos y la duración de cada sesión era de unos 30 minutos (incluyendo la preparación del experimento y el descanso).(Ali et al., 2021)

En el estudio de Kumar y compañía, consideraron unas señales EEG practicando patrones de agarre realizados por individuos sanos normales de acuerdo con las imágenes motores asociadas que se han monitorizado. El objetivo es desarrollar un marco conceptual para utilizar los datos recogidos y aplicar algoritmos de predicción adecuados que eviten el deslizamiento, con el fin de generar las fuerzas adecuadas en la punta de los dedos. Estas técnicas podrían utilizarse para el desarrollo de prótesis en el futuro. El experimento constó de individuos de entre 20 y 27 años. No tenían antecedentes de dolencias neuromusculares. Se pidió a los participantes que realizaran e imaginaran tareas de agarre con su brazo dominante. La secuencia de tareas se presentó de forma aleatoria a los sujetos. Los sujetos estaban sentados en una silla frente a una mesa. Seis objetos (una lata, una bola, un destornillador, un bolígrafo, una moneda y una tarjeta) se colocaron sobre la mesa en forma semicircular con un radio aproximado de 200 mm para una accesibilidad óptima con el codo pivotando en el centro.(Roy et al., 2018)

En este caso, las actividades de agarre implican estos cuatros dedos, hemos excluido

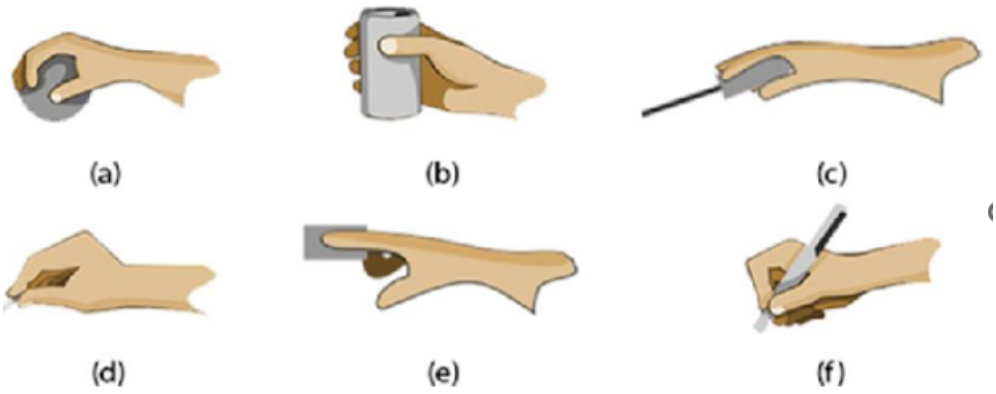

Figura 2

Diferentes tipos de patrón de agarre. (a). agarre esférico, (b). sujetador cilíndrico, (c). sujetador de herramientas eléctricas, (d). sujetador de pellizcos, (e). precisión lateral, (f). precisión del trípode.

Fuente: Roy et al., 2018

cualquier registro para el dedo meñique. Los sensores se conectaron a una placa Intel Galileo mediante un circuito divisor de tensión con una resistencia de 1MQ. Se registró el voltaje relacionado con la presión cutánea de la yema del dedo mientras se agarraba. El experimento se realizó en una sala insonorizada con dos sesiones diferentes. En la primera sesión, se pidió a los sujetos que realizaran el agarre real, mientras que en la segunda constó que imaginaran que agarraban los mismos objetos. Los sujetos mantuvieron un agarre firme de los objetos hasta que se presentó una señal auditiva de liberación.(Roy et al., 2018)

Para la tarea MI (imágenes motoras), lograron la intervención de seis sujetos humanos (edad promedio de 26,5 ± 3,5, con tres hombres). Cada sujeto se sentó frente a una

pantalla de computadora y se le indicó que realizará tareas de MI, incluida la ejecución del movimiento imaginario de la mano derecha o izquierda. Cada prueba se diseñó en cuatro pasos que incluyen fijación, señal, MI y descanso. Un icono de "descanso" apareció inicialmente en la pantalla durante 2 s después del tono de inicio. Esto guió a los sujetos a permanecer en un estado de reposo, es decir, a despejar su mente en preparación para la próxima tarea. Posteriormente, se presentó una señal izquierda o derecha durante 2 s, y el sujeto realiza continuamente la tarea de MI especificada durante 3,5 s, teniendo en cuenta el tiempo de reacción. Finalmente, se proporcionó un intervalo de 1 s como descanso adicional.(Roy et al., 2018)

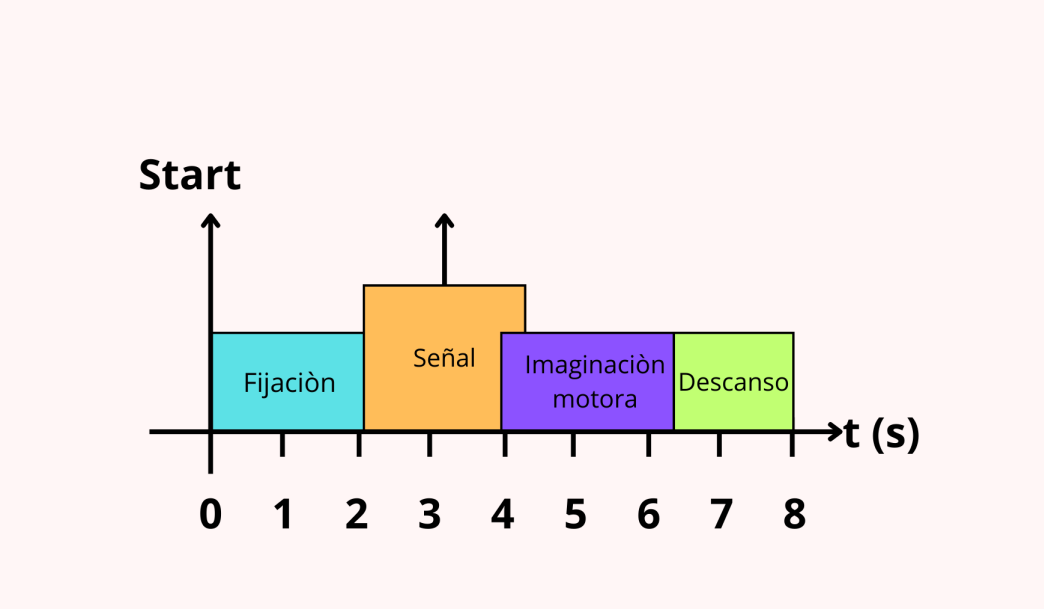

Figura 3

Pasos de la prueba para adquisición de señales

Fuente: Propia

2.2 Preprocesamiento de la señal

Es el primer paso del procesamiento de señales EEG y se utiliza para preparar los datos para su posterior análisis. Incluye la adquisición de señal, la eliminación de artefactos, el promediado de la señal, la umbralización de la salida y la mejora de la señal resultante. El objetivo de este paso es reducir el ruido y mejorar la calidad de la señal.

Filtración

Filtración de la señal: Las señales EEG sin procesar contienen ruido causado por factores externos que introducen ruido en la señal a través de artefactos, ejemplos de estos mencionados son los movimientos oculares, actividad muscular, respiración e incluso las redes eléctricas o magnéticas que estaban presentes cuando se realizó el experimento. Por lo tanto, se deben aplicar filtros adecuados para eliminar este tipo de ruido y obtener una señal clara y suficiente para su procesamiento.

La señal EEG sin procesar contiene algunas perturbaciones, las cuales se presentan debido a factores externos que provocan ruido en la señal. Esto se deben a señales provenientes de actividad muscular, movimientos oculares y actividad relacionada con el parpadeo, e incluso con la respiración o la red eléctrica. Por esta razón, es necesario aplicar filtros apropiados para eliminar este tipo de ruidos y obtener una señal nítida y adecuada para poder trabajar sobre ella.(Granata et al., 2020). En la siguiente tabla, se encuentran los filtros usados por algunos investigadores que se tomaron como referencia:

FILTRO	FRECUENCIA (Hz)	REFERENCIA
Paso de banda	0.1 - 50	(Alazrai et al., 2019),(Granata et al., 2020),(Roy et al., 2020)
Filtro de respuesta de impulso finito	(Tiempo finito a cero)	(Granata et al., 2020),(Yang et al., 2020)
Pasa banda	1 - 80	(Granata et al., 2020)
Filtro de muesca	50	(Granata et al., 2020),(Yang et al., 2020)
La caja de herramientas EEGLAB de la interfaz de Matlab	-	(Ali et al., 2021),(Roy et al., 2020),(Zhang et al., 2019)
Paso de banda Butterworth de 8° orden	0.01 - 60	(Yang et al., 2020),(Zhang et al., 2019)
Filtrado laplaciano	(De realce)	(Yang et al., 2020)
Paso de banda Butterworth de 6° orden hacia adelante y hacia atrás	1 - 12	(Roy et al., 2018)

Cuadro 2

Filtros que más se implementan. Fuente: Propia

El filtrado es un paso casi omnipresente en el preprocesamiento de datos de EEG y MEG. El filtrado podría cambiar seriamente la apariencia del señales y por lo tanto afectan los resultados obtenidos. Ciertas investigaciones sugieren no filtrar en absoluto o filtrar con mucha precaución.

Se ha observado que se utiliza en diversas investigaciones el filtrado pasa banda, el cual posee ciertos parámetros como el orden de filtro, lo cual afecta en gran medida la señal filtrada resultante.

La elección de los órdenes de filtro α y φ corresponde fuertemente a pendientes del filtro en la característica de Bode de la función de transferencia del filtro. Generalmente, el filtro de paso de banda no entero tiene pendientes asimétricas. En particular para bajas frecuencias tiene una pendiente ascendente de 20φdB/dec mientras que para frecuencias más altas tiene una pendiente descendente de $20(2\alpha - \varphi)$ dB/dec.(Baranowski et al., 2016)

En general, el diseño de filtros para señales EEG es difícil. Se recomiendan ampliamente los filtros menos profundos ya que producen menos distorsiones de la señal y difundirlas menos en el dominio del tiempo.(Baranowski et al., 2016)

Extracción de caracteristicas

Este paso consiste en seleccionar y calcular características útiles de las señales EEG. Las características se utilizan para describir las propiedades de las señales y pueden ser de diferentes tipos, como estadísticas, tiempo-frecuencia, topográficas, entre otras. La selección adecuada de características es crucial para el éxito del análisis y la clasificación de las señales EEG.

Para iniciar en la extracción de características, es necesario normalizar o adecuar el conjunto de parámetros con el objetivo de que sea posible trabajar sobre ellos para finalmente clasificarlos. Por ello, se utilizan diferentes métodos para identificar las variables independientes en cada uno de los modelos y así el programa sea capaz de aprender de estos datos, identificar los patrones y tomar decisiones con mínima intervención humana.

- Extracción de características dominio tiempo:

Las principales características que se pueden obtener en el dominio del tiempo son características probabilísticas o propias de las señales.

Valor medio absoluto(MAV), es el promedio de los valores absolutos de las N muestras tomadas.(Sanabria Hernández et al., s.f.)

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{1}$$

Varianza(VAR), define la dispersión de los datos con relación a la media(Sanabria Hernández et al., s.f.)

$$VAR = \frac{1}{N-1}\sum_{i=1}^{N}x_i^2 \tag{2}$$

Valor eficaz (RMS), es un valor escalar, correspondiente a la media cuadrática, es usado porque ayuda a obtener la amplitud de la señal en el movimiento muscular tomado en sEMG.(Sanabria Hernández et al., s.f.)

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2} \tag{3}$$

Integral cuadrada simple (SSI), es la suma de los valores cuadrados de la señal en segmento de tiempo, como se indica en la ecuación.(Sanabria Hernández et al., s.f.)

$$SSI = \sum_{i=1}^{N}xi^2 \tag{4}$$

Longitud de onda (WL), Se puede calcular simplificando la longitud acumulada de la suma de formas de onda. Se puede definir como:(Too et al., 2019)

$$WL = \sum_{i=1}^{N-1}|x_{i+1} - x_i| \tag{5}$$

Cambio de pendiente (SSC), Número de veces que la pendiente de la señal cambia de signo.(Sanabria Hernández et al., s.f.)

Parámetros de Hjorth, son tres parámetros ampliamente usados en EEG. El primero es la actividad, el cual representa la varianza de una señal en tiempo frecuencial, el siguiente parámetro es la movilidad, que representa la proporción de la desviación estándar del espectro y por último la complejidad que representa la similitud de la onda con una onda sinusoidal.(Sanabria Hernández et al., s.f.)

$$actividad(x(t)) = VAR(x(t)) \tag{6}$$

$$movilidad(x(t)) = \sqrt{\frac{actividad(x'(t))}{actividad(x(t))}} \tag{7}$$

$$complejidad(x(t)) = \frac{movilidad(x'(t))}{movilidad(x(t))} \tag{8}$$

Exponente de Hjorth, es un análisis estadístico utilizado para hallar la correlación de datos, se define como la relación de los máximos valores de los datos normalizado con la desviación estándar.(Sanabria Hernández et al., s.f.)

$$\frac{R(s)}{S(n)} = \frac{1}{\sqrt{S^2(n)}} \tag{9}$$

$(max(0, W_1, W_2, W_3, ..., W_n) - min(0, W_1, W_2, W_3, ..., W_n))$

Dimensión Fractal (PFD), hace referencia formas o fluctuaciones en el tiempo que guardan auto-similaridades. Como métrica, la dimensión fractal nos indica el grado de complejidad de la señal analizada.(Aldás et al., 2018)

$$FD_{Petrosian} = \frac{log_{10}n}{log_{10}n + log_{10}(\frac{n}{n+0{,}4N_{\triangle}})} \tag{10}$$

Variación media de la amplitud (AAC), El cambio de amplitud promedio (AAC) es casi equivalente a la función WL, excepto que se promedia la longitud de onda. Se puede formular como:(Phinyomark et al., 2012)

$$AAC = \frac{1}{N}\sum_{i=1}^{N-1}|x(i+1) - xi| \tag{11}$$

Diferencia Desviación Estándar Absoluta Valor (DASDV), El valor de desviación estándar absoluta de diferencia (DASDV) se parece a la función RMS, en otras palabras, es un valor de desviación estándar de la longitud de onda, como se puede definir por:(Phinyomark et al., 2012)

$$DASDV = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1}(x(i+1) - xi)^2} \tag{12}$$

Valor absoluto medio modificado (MMAV), El valor absoluto medio modificado tipo 1 (MAV1) es una extensión de la función MAV. La función de ventana ponderada wi se asigna a la ecuación para mejorar la robustez de la función MAV. Se calcula por:(Phinyomark et al., 2012)

$$MMAV = \frac{1}{N}\sum_{i=1}^{N}(wi|xi|) \tag{13}$$

$$\text{Con } wi = \begin{cases} 1, if & 0{,}25 \leq i \leq 0{,}75N \\ 0{,}5, & otherwise \end{cases}$$

Valor absoluto medio modificado 2 (MMAV2), El valor absoluto medio modificado tipo 2 (MAV2) es una expansión de la función MAV que es similar al MAV1. Sin embargo, la función de ventana ponderada wi que se asigna a la ecuación es una función continua. Mejora la suavidad de la función ponderada. La ecuación se

define como:(Phinyomark et al., 2012)

$$MMAV2 = \frac{1}{N}\sum_{i=1}^{N}(wi|xi|) \tag{14}$$

$$\text{Con } wi = \begin{Bmatrix} 1, if & 0{,}25 \leq i \leq 0{,}75N \\ 4i/N, elseif & i < 0{,}25N \\ 4(i-N)/N, & otherwise \end{Bmatrix}$$

Longitud máxima fractal (MFL), Los autores han identificado la longitud máxima del fractal (MFL) como una medida de la fuerza de contracción. Si bien es similar a la longitud de onda, MFL incorpora la escala logarítmica, lo que la hace menos sensible al ruido. Por lo tanto, pueden identificar con precisión acciones con múltiples músculos y contracciones de bajo nivel. Para calcular el MFL requiere el cálculo de la longitud de la curva, Xk^m, para una señal de tiempo muestreada a una tasa de muestreo fija, x(n) = X(1), X(2), X(3), ..., X(N) como sigue:(Arjunan y Kumar, 2010)

$$Lm(k) = \frac{(\sum_{i=1}^{\frac{N-mk}{k}} |X(m+ik) - X(m+(i-1)k)|)(\frac{N-1}{\frac{k(N-m)}{k}})}{k} \tag{15}$$

Detector de log (LD), Al igual que la función vOrder, esta función también proporciona una estimación de la fuerza muscular ejercida. El detector no lineal se caracteriza como log($|Xk|$) y la función logDetect se define como:(Tkach et al., 2010)

$$LD = e^{\frac{1}{N}\sum_{i=1}^{N}(log(|Xk|)} \tag{16}$$

Función de dimensión de correlación (CD), Se puede calcular directamente a partir de la serie temporal utilizando el algoritmo GP propuesto por Grassberger y Procaccia. Este algoritmo se basa en el teorema de incrustación y la construcción del

espacio de fase y se ha convertido en el enfoque más utilizado para estimar las dimensiones fractales de conjuntos de datos experimentales. Se puede estimar por:(Wang et al., 2014)

$$Cd = \lim_{r \to 0} \frac{\delta ln C_m(r)}{\delta ln r} \tag{17}$$

Valor Absoluto Medio Mejorado (EMAV)(Too et al., 2019)

$$EMAV = \frac{1}{N} \sum_{i=1}^{N} |(x_i)^p| \tag{18}$$

$$\text{Con } p = \left\{ \begin{matrix} 0{,}75, if & i \geq 0{,}2N & and & i \leq 0{,}8N \\ 0{,}5, & otherwise & & \end{matrix} \right\}$$

Longitud de onda mejorado (EWL)(Too et al., 2019)

$$WL = \sum_{i=2}^{N} |(x_{i+1} - x_i)^p| \tag{19}$$

$$\text{Con } p = \left\{ \begin{matrix} 0{,}75, if & i \geq 0{,}2N & and & i \leq 0{,}8N \\ 0{,}5, & otherwise & & \end{matrix} \right\}$$

Covarianza y correlación de datos

Un coeficiente de correlación mide el grado en que dos variables tienden a cambiar al mismo tiempo. El coeficiente describe tanto la fuerza como la dirección de la relación. Existen dos análisis de correlación diferentes: ("Correlación y Covarianza", s.f.) Correlación del momento del producto de Pearson: La correlación de Pearson evalúa la relación lineal entre dos variables continuas. Una relación es lineal cuando un cambio en una variable se asocia con un cambio proporcional en la otra variable.("Correlación y Covarianza", s.f.) Se representa mediante el cociente entre la

covarianza de las variables y la raíz cuadrada del producto de la varianza de cada variable:

$$CorrP(x,y) = \frac{Cov(x,y)}{\sqrt{Var(x).Var(y)}} \tag{20}$$

Correlación del orden de los rangos de Spearman:La correlación de Spearman evalúa la relación monótona entre dos variables continuas u ordinales. En una relación monótona, las variables tienden a cambiar al mismo tiempo, pero no necesariamente a un ritmo constante. El coeficiente de correlación de Spearman se basa en los valores jerarquizados de cada variable y no en los datos sin procesar.Se representa mediante:

$$CorrR_s = 1 - \frac{6\sum(di)^2}{n(n^2-1)} \tag{21}$$

Donde di es la diferencia de rangos entre variables x y y y n el número total de datos. Siempre es una buena idea examinar la relación entre las variables con una gráfica de dispersión. Los coeficientes de correlación solo miden relaciones lineales (Pearson) o monótonas (Spearman). Son posibles otras relaciones.(“Correlación y Covarianza”, s.f.)

La covarianza de diversos datos mide la relación lineal entre dos variables mediante una matriz cuadrada n*m. Se expresa mediante la siguiente ecuación:

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(xi.yi)}{n} - \bar{x}\bar{y} \tag{22}$$

Aunque la covarianza es similar a la correlación entre variables, difieren de las siguientes maneras:

Los coeficientes de correlación están estandarizados. Por lo tanto, una relación lineal perfecta da como resultado un coeficiente de 1. La correlación mide tanto la fuerza como la dirección de la relación lineal entre dos variables.(“Correlación y Covarianza”, s.f.)

Los valores de covarianza no están estandarizados. Por consiguiente, la covarianza puede ir desde infinito negativo hasta infinito positivo. Por lo tanto, el valor de una relación lineal perfecta depende de los datos. Puesto que los datos no están estandarizados, es difícil determinar la fuerza de la relación entre las variables.(“Correlación y Covarianza”, s.f.)

Se puede utilizar la covarianza para comprender la dirección de la relación entre las variables. Los valores de covarianza positivos indican que los valores por encima del promedio de una variable están asociados con los valores por encima del promedio de la otra variable y los valores por debajo del promedio están asociados de manera similar (relación directamente proporcional). Los valores de covarianza negativos indican que los valores por encima del promedio de una variable están asociados con los valores por debajo del promedio de la otra variable (relación inversamente proporcional).(“Correlación y Covarianza”, s.f.)

El coeficiente de correlación depende de la covarianza. El coeficiente de correlación es igual a la covarianza dividida entre el producto de las desviaciones estándar de las variables. Por lo tanto, una covarianza positiva siempre producirá una correlación positiva y una covarianza negativa siempre generará una correlación negativa.(“Correlación y Covarianza”, s.f.)

Análisis de componentes principales (PCA)

El análisis de componentes principales (PCA) se ha utilizado ampliamente en varios campos de investigación para reducir la dimensionalidad del espacio del sensor original y simplificar los análisis posteriores. Por medio de una rotación ortogonal, el PCA transforma linealmente un conjunto de canales de datos de entrada en un número igual de variables no correlacionadas linealmente (componentes principales, PC) que representan sucesivamente la mayor parte posible de la varianza de datos restante.(Artoni et al., 2018)

El PCA puede servir tanto como una herramienta de análisis exploratorio como para proporcionar una visualización e interpretación simplificadas de un conjuntos de datos multivariados.(Artoni et al., 2018)

El PCA se calcula como la descomposición de valores propios de la matriz de covarianza $C_x = XX^T$. La parte de la varianza de los datos representan los primeros p componentes, como una relación porcentual con respecto al total de varianza del conjunto de datos.(Artoni et al., 2018)

2.3 Clasificación

En este paso, se utiliza una técnica de aprendizaje automático para clasificar las señales EEG en diferentes categorías o clases. Se pueden utilizar diferentes algoritmos de clasificación, como análisis lineal, análisis no lineal, algoritmos adaptativos, clustering y fuzzy. La elección del algoritmo de clasificación depende de la naturaleza de los datos y del objetivo de la clasificación.

Existen diversos métodos de clasificación, sin embargo el proyecto de investigación tiene por objetivo emplear como clasificador la máquina de vector de soportes

(SVM), uno de los más usados para métodos como el expuesto anteriormente, ya que ha resultado pertinente para un método de extracción de características que favorezca una categorización multiclase.

La máquina de vector de soportes es una nueva técnica de clasificación y ha tomado mucha atención en años recientes. La teoría de la SVM esta basada en la idea de minimización de riego estructural (SRM). En muchas aplicaciones, las SVM han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación.(Betancourt, 2005)

Una SVM primero mapea los puntos de entrada a un espacio de características de una dimensión mayor (i.e.: si los puntos de entrada están en $\Re^2$ entonces son mapeados por la SVM a $\Re^3$) y encuentra un hyperplano que los separe y maximice el margen m entre las clases en este espacio como se aprecia en la Figura 4(Betancourt, 2005).

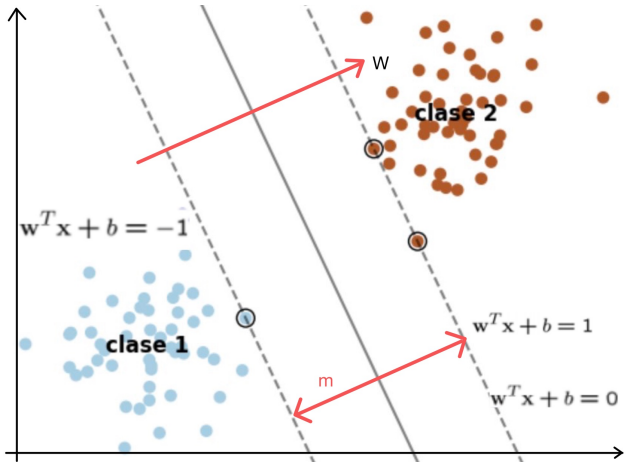

Figura 4

La frontera de decisión debe estar tan lejos de los datos de ambas clases como sea. Fuente: Propia

Maximizar el margen m es un problema de programación cuadrática (QP) y puede ser resuelto por su problema dual introduciendo multiplicadores de Lagrange. Sin ningún conocimiento del mapeo, la SVM encuentra el hyperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas kernels. La solución del hyperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son llamados vectores de soporte.(Betancourt, 2005)

Caso linealmente separable. Supongamos que nos han dado un conjunto S de puntos etiquetados para entrenamiento como se aprecia en la Figura 5.

$$(y1, x1), ..., (yi, xi) \tag{23}$$

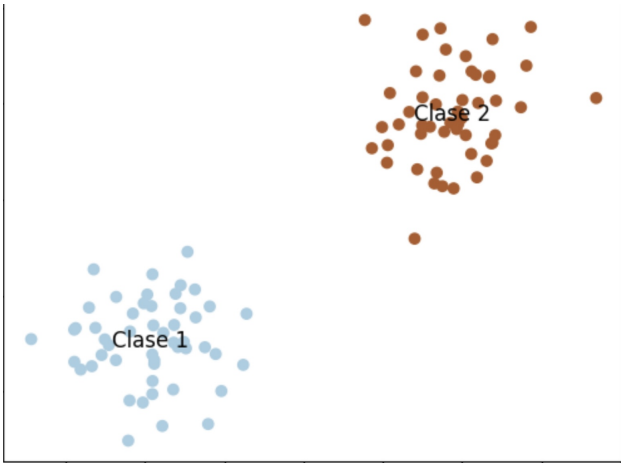

Figura 5

Caso linealmente separable. Fuente: Propia

Cada punto de entrenamiento $x_i \, \epsilon \, \Re^N$ pertenece a alguna de dos clases y se le ha dado una etiqueta $y_i \, \epsilon \, (-1, 1)$ para $i = l, ..., l$. En la mayoría de los casos, la búsqueda

de un hyperplano adecuado en un espacio de entrada es demasiado restrictivo para ser de uso práctico. Una solución a esta situación es mapear el espacio de entrada en un espacio de características de una dimensión mayor y buscar el hyperplano óptimo allí. Sea $z = \varphi(x)$ la notación del correspondiente vector en el espacio de características con un mapeo φ de $\Re^N$ a un espacio de características Z. Deseamos encontrar el hyperplano.(Betancourt, 2005)

$$w \cdot z + b = 0 \tag{24}$$

Definido por el par (w, b), tal que podamos separar el punto xi de acuerdo a la función

$$f(x_i) = sign(w \cdot z_i + b) = \begin{Bmatrix} 1 & y_i = 1 \\ -1 & y_i = -1 \end{Bmatrix} \tag{25}$$

Donde $w \epsilon Z$ y $b \epsilon \Re$. Más precisamente, el conjunto S se dice que es linealmente separable si existe (w, b) tal que las inecuaciones

$$\begin{Bmatrix} (w \cdot z_i + b) \geq 1, & y_i = 1 \\ (w \cdot z_i + b) \leq -1 & y_i = -1 \end{Bmatrix} i = 1, ..., l \tag{26}$$

Sean válidas para todos los elementos del conjunto S. Para el caso linealmente separable de S, podemos encontrar un único hyperplano óptimo, para el cual, el margen entre las proyecciones de los puntos de entrenamiento de dos diferentes clases es maximizado.(Betancourt, 2005)

Caso no linealmente separable. Si el conjunto S no es linealmente separable, violaciones a la clasificación deben ser permitidas en la formulación de la SVM.(Betancourt, 2005)

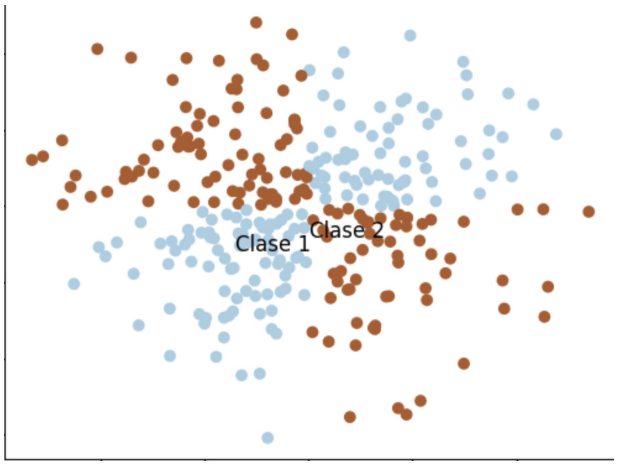

Figura 6

Caso no linealmente separable. Fuente: Propia

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no-negativas $\xi_i \geq 0$ de tal modo que (26) es modificado a: (Betancourt, 2005)

$$y_i(w \cdot z_i + b) \geq 1 - \xi_i, \qquad i = 1, ...l. \tag{27}$$

Los $\xi_i \neq 0$ (27) son aquellos para los cuales el punto x_i no satisface para el caso linealmente separable. Entonces el término $\sum_{i=1}^{l} \xi_i$ puede ser tomado como algún tipo de medida del error en la clasificación.(Betancourt, 2005)

El problema del hyperplano óptimo es entonces redefinido como la solución al problema(Betancourt, 2005)

$$min \qquad \{\frac{1}{2} w \cdot w + C \sum_{i=1}^{l} \xi_i\} \tag{28}$$

$$s.a \quad y_i(w \cdot z_i + b) \geq 1 - \xi_i,$$

$$i = 1, ..., l \; \xi_i \geq 0, \quad i = 1, ..., l$$

Donde C es una constante. El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de éste parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación.(Betancourt, 2005)

Buscando el hyperplano óptimo en (28) es un problema QP, que puede ser resuelto construyendo un Lagrangiano y transformándolo en el dual .(Betancourt, 2005)

$$Max \quad W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j z_i \cdot z_j \tag{29}$$

$$s.a \quad \sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., l \tag{30}$$

Donde $\alpha = (\alpha_1, ..., \alpha_l)$ es un vector de multiplicadores de Lagrange positivos asociados con las constantes en (30).

El teorema de Khun-Tucker juega un papel importante en la teoría de las SVM. De acuerdo a este teorema, la solución α_i del problema (30) satisface: (Betancourt, 2005)

$$\overline{\alpha_i}(\overline{w} \cdot z_i + \overline{b}) - 1 + \overline{\xi_i}) = 0, \quad i = 1, ..., l \tag{31}$$

$$(C - \overline{\alpha_i})\xi_i = 0, \quad i = 1, ...l \tag{32}$$

De esta igualdad se deduce que los únicos valores $\overline{\alpha_i} \neq 0$ (32) son aquellos que para las constantes (30) son satisfechas con el signo de igualdad. El punto xi correspondiente con $\overline{\alpha_i} > 0$ es llamado vector de soporte, pero hay dos tipos de vectores de soporte en un caso no separable. En el caso $0 < \overline{\alpha_i} > C$, el correspondiente vector de soporte x_i satisface las igualdades $y_i(\overline{w} \cdot z_i + \overline{b}) - 1$ y $\xi_i = 0$. En el caso $\overline{\alpha_i} = C$, el correspondiente ξ_i es diferente de cero y el correspondiente vector de soporte x_i no satisface en el caso linealmente separable. Nos referimos a

estos vectores de soporte como errores. El punto x_i corresponde con $\overline{\alpha_i} = 0$ es clasificado correctamente y esta claramente alejado del margen de decisión.(Betancourt, 2005)

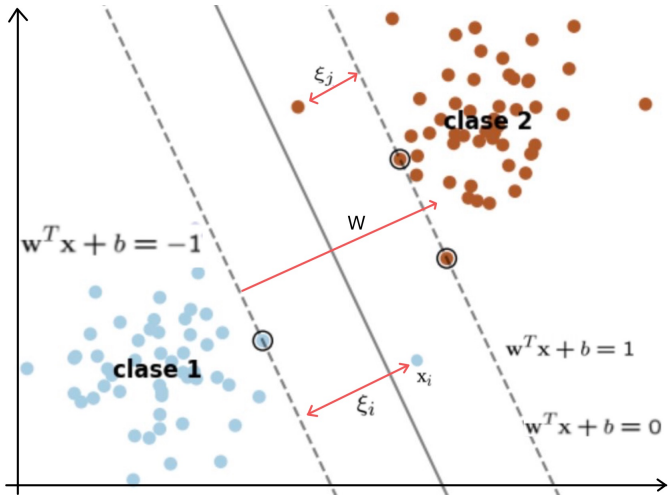

Figura 7

Aparición del parámetro de error ξ_i en el error de clasificación. Fuente: Propia

Para construir el hyperplano óptimo $\overline{w} \cdot z + \bar{b}$, se utiliza

$$\overline{w} = \sum_{i=1}^{l} \overline{\alpha_i} y_i z_i \tag{33}$$

Y el escalar b puede ser determinado de las condiciones de Kuhn-Tucker. La función de decisión generalizada de (28) y (33) es tal que

$$f(x) = sign(w \cdot z + b) = sign(\sum_{i=1}^{l} \overline{\alpha_i} y_i z_i \cdot z + b) \tag{34}$$

Truco del Kernel para el caso no linealmente separable. Como no tenemos ningún conocimiento de φ , el cálculo del problema (32) y (34) es imposible. Hay una buena propiedad de la SVM la cual es que no es necesario tener ningún conocimiento

acerca de φ . Nosotros sólo necesitamos una función K($\cdot$, $\cdot$) llamada kernel que calcule el producto punto de los puntos de entrada en el espacio de características Z, esto es (Betancourt, 2005)

$$z_i \cdot z_j = \varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j) \tag{35}$$

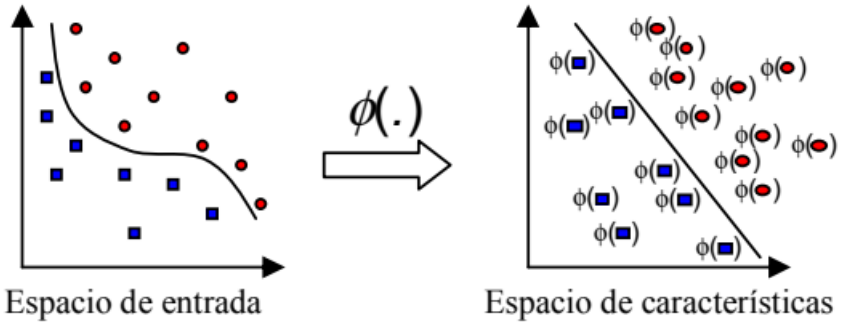

Figura 8

Idea del uso de un kernel para transformación del espacio de los datos. Fuente:Betancourt, 2005

Las funciones que satisfacen el teorema de Mercer pueden ser usadas como productos punto y por ende pueden ser usadas como kernels. Podemos usar el kernel polinomial de grado d (Betancourt, 2005)

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^d \tag{36}$$

Para construir un clasificador SVM:

Entonces el hyperplano no lineal de separación puede ser encontrado como la solución de

$$Max \quad W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{37}$$

$$s.a \quad \sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., l \tag{38}$$

y la función de decisión es

$$f(x) = sign(w \cdot z + b) = sign(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x_j) + b) \tag{39}$$

Hardware y Software OpenBCI

Una interfaz como la de OpenBCI proporciona libertad y flexibilidad incomparables a través de hardware y firmware de código abierto con una implementación de bajo costo. Explota plataformas de hardware robustas y potentes kits de desarrollo de software para crear controladores personalizados con capacidades avanzadas. (Cardona-Álvarez et al., 2023)

Un sistema OpenBCI bien estructurada, puede ofrecer una retroalimentación de señal en tiempo real y una ejecución controlada de protocolos clínicos para los registros neuronales y hasta su análisis en tiempo real.

Aún así, varias restricciones pueden reducir significativamente el rendimiento de OpenBCI. Estas limitaciones incluyen la necesidad de una comunicación más efectiva entre computadoras y dispositivos periféricos y más flexibilidad para configuraciones rápidas bajo protocolos específicos para datos neurofisiológicos. (Cardona-Álvarez et al., 2023)

A continuación se describen algunos fundamentos de componentes de hardware y software del OpenBCI.

OpenBCI es una opción de hardware de código abierto altamente flexible para aplicaciones de biodetección. La placa puede funcionar con señales EEG, admite electromiografía (EMG) y electrocardiografía. Además, la placa de biosensor Cyton, cuenta con un microcontrolador PIC32MX250F128B, un gestor de arranque ChipKIT UDB32-MX2-DIP, un acelerómetro de 3 ejes LIS3DH y un convertidor analógico a digital ADS1299 con ocho canales de entrada (ampliables a 16) con una frecuencia de muestreo máxima de 16 kHz.(Cardona-Álvarez et al., 2023)

Figura 9

Auricular EEG "mark IV". Tomada de "OpenBCI/Ultracórtex Mark IV", s.f.

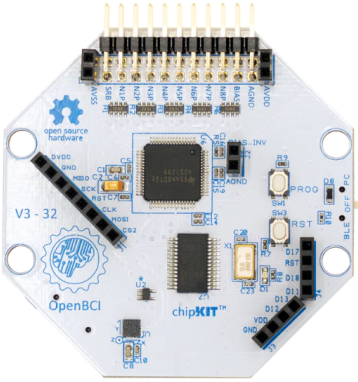

Figura 10

Tablero de cyton. Tomada de "OpenBCI/Tablero de ganglios", s.f.

Los canales de EEG se pueden configurar en modo monopolar o bipolar, con la capacidad de agregar hasta cinco entradas digitales externas y tres entradas analógicas. El Protocolo de control de transmisión (TCP) también puede acceder al flujo de datos a través de una interfaz Wi-Fi.(Cardona-Álvarez et al., 2023)

Otra opción es RFduino, que por defecto admite 250 muestras por segundo (SPS) y ocho canales. Sin embargo, con la adición del módulo Daisy, se puede ampliar a 16 canales y con el escudo Wi-Fi, la frecuencia de muestreo puede aumentar a 16 kHz. Es de destacar que todo el montaje de electrodos se puede configurar en modos monopolar, bipolar o secuencial.(Cardona-Álvarez et al., 2023)

Figura 11

Wifi Shield. Tomada de "Wifi shield", s.f.

Figura 12

Tablero Cyton + Daisy. Tomada de "OpenBCI/Tablero de ganglios", s.f.

La siguiente tabla resume las principales configuraciones de OpenBCI:

OpenBCI Cyton	Channels	Digital Inputs	Analog Inputs	Max. Sample Rate	Featured Protocol
RFduino	8	5	3	250 Hz	Serial
RFduino + Daisy	16	5	3	250 Hz	Serial
RFduino + Wi-Fi shield	8	2	1	16 KHz	TCP (over Wi-Fi)
RFduino + Wi-Fi shield + Daisy	16	2	1	8 KHz	TCP (over Wi-Fi)

Cuadro 3

Configuraciones OpenBCI Cyton usando placa de expansión Daisy y escudo Wi-Fi.

Fuente: Cardona-Álvarez et al., 2023

2.4 Evaluación de modelos de clasificación para EEG

La evaluación de modelos de clasificación para EEG es una tarea importante para garantizar que los modelos sean precisos y confiables. Las métricas de evaluación se utilizan para cuantificar el desempeño de un modelo de clasificación y compararlo con otros modelos.

2.4.1 Matriz de confusión

La matriz de confusión es una tabla que se utiliza para visualizar el desempeño de un algoritmo de clasificación. Esta tabla se construye a partir de los resultados de la clasificación de un conjunto de datos de prueba.

La matriz de confusión consta de cuatro celdas, que se representan de la siguiente

manera:

1. **Verdaderos positivos (TP)**: Son los casos en los que el algoritmo de clasificación predice correctamente la clase real del ejemplo.
2. **Falsos positivos (FP)**: Son los casos en los que el algoritmo de clasificación predice erróneamente la clase "positivo"para un ejemplo que en realidad pertenece a la clase "negativo".
3. **Verdaderos negativos (TN)**: Son los casos en los que el algoritmo de clasificación predice correctamente la clase real del ejemplo.
4. **Falsos negativos (FN)**: Son los casos en los que el algoritmo de clasificación predice erróneamente la clase "negativo"para un ejemplo que en realidad pertenece a la clase "positivo".

2.4.2 Metricas de evaluación

Las métricas de evaluación más comunes para clasificación de EEG incluyen:

1. **Exactitud:** La exactitud es la proporción de casos en los que el modelo de clasificación predice correctamente la clase real del ejemplo. Se calcula como:

$$Exactitud = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{40}$$

donde:

a) **TP**: Verdaderos positivos

b) **TN**: Verdaderos negativos

c) **FP**: Falsos positivos

Figura 13

MATRIZ DE CONFUSIÓN BINARIA. Tomada de Barrios Arce, 2019

d) **FN**: Falsos negativos

2. **Precisión**: La precisión es la proporción de casos en los que el modelo de clasificación predice la clase "positivo"que realmente pertenecen a la clase "positivo". Se calcula como:

$$Precisión = \frac{TP}{(TP + FP)} \tag{41}$$

3. **Recall**: El recall es la proporción de casos en los que el modelo de clasificación predice la clase "positivo"que realmente pertenecen a la clase "positivo". Se calcula como:

$$Recall = \frac{TP}{(TP + FN)} \tag{42}$$

4. **F1-score**: El F1-score es una métrica que combina la precisión y el recall. Se calcula como:

$$F1 - score = 2 * \frac{(Precisión * Recall)}{(Precisión + Recall)} \tag{43}$$

Estas métricas se pueden utilizar para comparar el desempeño de diferentes modelos de clasificación para EEG(Cai y et al., 2021). Por ejemplo, si un modelo tiene una exactitud del 90 % y un recall del 95 %, entonces este modelo es mejor que un modelo que tiene una exactitud del 80 % y un recall del 90 %.

2.4.3 Curva ROC

La curva ROC (Receiver Operating Characteristic) es una herramienta gráfica que se utiliza para evaluar el desempeño de un modelo de clasificación binario. La curva ROC representa la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) para diferentes valores del umbral de clasificación.

La TPR es la proporción de casos positivos que el modelo predice correctamente como positivos. La FPR es la proporción de casos negativos que el modelo predice erróneamente como positivos.

Una curva ROC ideal se desplazaría hacia la esquina superior izquierda del gráfico, lo que indicaría que el modelo tiene un buen desempeño en la clasificación de ambos tipos de casos(Fawcett, 2006).

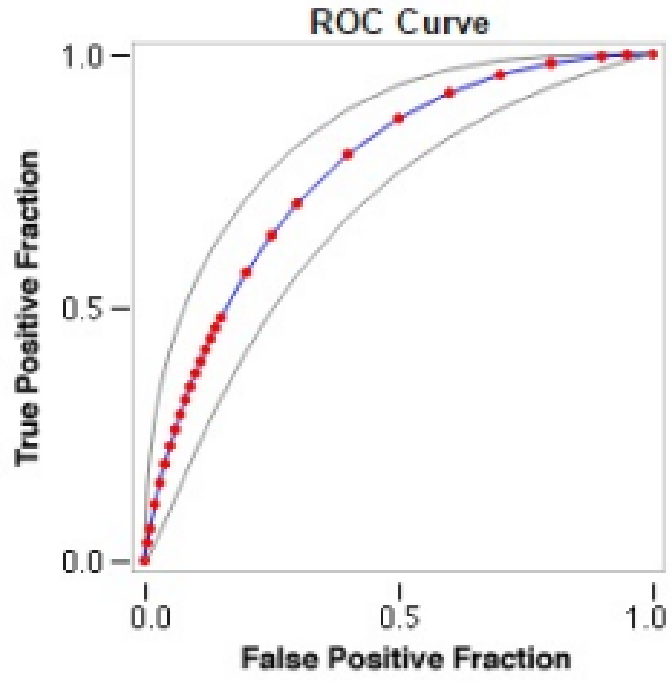

Figura 14

Curva de ROC. Tomada de Eng, s.f.

2.4.4 Área bajo la curva ROC (AUC)

El área bajo la curva ROC (AUC) es una métrica que se utiliza para cuantificar el desempeño de un modelo de clasificación binario(Fawcett, 2006). El AUC es un valor entre 0 y 1, donde 0 indica un desempeño muy pobre y 1 indica un desempeño perfecto.

Un AUC alto indica que el modelo es capaz de distinguir entre las dos clases con precisión.

$$AUC = \int_0^1 TPR(FPR)dFPR \tag{44}$$

donde:

1. **TPR**: Tasa de verdaderos positivos
2. **FPR**: Tasa de falsos positivos

Sección 3

3. METODOLOGIA

Para el cumplimiento del proyecto, se debe seguir la siguiente metodología:

3.1 Revisión bibliográfica

Se realizará una determinada profundización en los temas propuestos mediante la recopilación de artículos referentes y similares según el campo de estudio a tratar, que consta del procedimiento y tipo de señales obtenidas (machine learning y señales EEG).

3.2 Recolección y creación de la base de datos

Se colocaron ocho electrodos en los lados frontal, temporal, central, occipital y dos electrodos en cada oreja (Fp1, Fp2, T5, T6, C3, C4, O1, O2, A1 y A2) según el 10–20 sistema de colocación de EEG (Fig. 15); El sistema de colocación EEG 10/20 es un método reconocido internacionalmente donde los electrodos son colocados en el cuero cabelludo del individuo, utilizando la proporción de 10 o 20 % de la distancia total entre los puntos de referencia anatómicos de la cabeza. (Cantarelli et al., 2016) La ubicación correspondiente de la cabeza está representada por el número de posiciones dadas. Los números pares indican el lado derecho de la cabeza, mientras que los números impares indican el lado izquierdo. Los datos EEG se obtuvieron utilizando una placa Cyton OpenBCI y el software OpenBCI GUI para una estática de 3 minutos medición a 30 fps, muesca de 60 Hz y filtro de PA de 1 a 50 Hz. Los datos se muestrean a 250 Hz en cada uno de los ocho canales.

La ubicación de los ocho electrodos mencionados anteriormente, permiten una gran cobertura en la corteza cerebral, sin embargo, se optó por posicionarlos así por las funciones de cada lóbulo cerebral. En el caso del electrodo C3 y C4, son las posiciones más usadas en proyectos de imaginación motriz porque se encuentran en el área de la función motora. Los electrodos en la parte occipital permiten recolectar la información visual, ya que existe relación entre lo que observa el paciente y la acción de movimiento durante el experimento, asimismo en el caso de los electrodos ubicados en el lóbulo frontal que abren paso al control y planificación, y los electrodos ubicados en el lóbulo temporal en cuanto a la memoria.

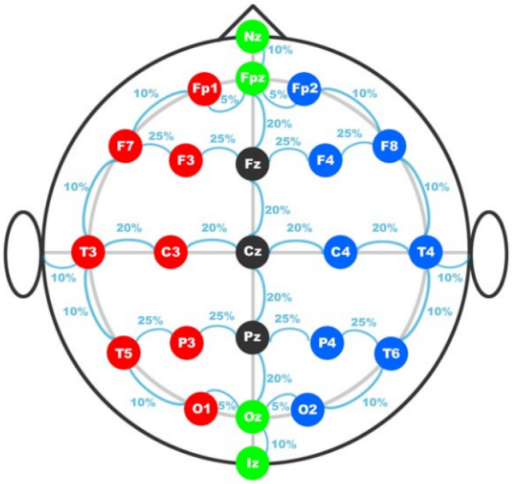

Figura 15

Sistema 10-20 de colocación EEG.

Tomada de Espinosa, 2016

Para la obtención de las señales EEG se tomó como base experiencias en investigación que describen el procedimiento para la recolección de datos. En primer

lugar, se seleccionaron personas sin limitaciones físicas, cognitivas, que no presenten abusos de algún tipo de medicamento o sustancia psicoactiva. Seguidamente, se obtienen las señales EEG al mismo tiempo que la persona sigue algunas instrucciones. El sitio de la realizaciòn del protocolo fue un espacio cerrado e insonoro para evitar las perturbaciones auditivas.

El grupo participante comprende las edades entre 19 años y 54 años, como se muestra en la siguiente tabla.

ID	Sexo	Edad	Epilepsia	Trastornos Neurológicos	Lesiones Cerebrales	Otras Condiciones Médicas
1	M	19	No	Migrañas	No	Ninguna
2	M	54	No	No	No	Ninguna
3	F	48	No	No	No	Trastorno de ansiedad
4	M	21	No	No	No	Ninguna
5	M	21	No	No	No	Ninguna
6	M	25	No	No	No	Asma
7	F	23	No	No	No	Ninguna
8	F	21	No	No	No	Depresión
9	M	21	No	No	No	Ninguna

Cuadro 4

Información general de los participantes en el conjunto de datos.

Fuente: Propia

Antes de realizar la recolección de señales electroencefalográficas, a cada paciente se le informó y se le sugirió acerca de unas recomendaciones para garantizar la precisión y efectividad en las pruebas de recolección. Las sugerencias impartidas fueron las siguientes:

1. **Limpieza del cabello**: Evitar utilizar acondicionador, cremas o productos capilares antes de la toma de señal, además de realizar una limpieza exhaustiva del cuero cabelludo, ya que un cabello limpio facilita el contacto de los electrodos.
2. **Cafeína u otros estimulantes**: Abstener el consumo de alimentos o bebidas que contengan cafeína al menos 8 horas antes de la prueba.
3. **Descanso adecuado**: Descansar lo suficiente la noche anterior a la prueba para minimizar la fatiga y garantizar una actividad cerebral más natural.
4. **Consumo de cigarrillo, alcohol y/o sustancias psicoactivas**: Evitar el consumo de estas sustancias por lo menos 48 horas antes de la prueba, ya que afectan la actividad cerebral.
5. **Relaciones sexuales o actividad física intensa**: Abstener de realizar actividades físicas intensas o relaciones sexuales por lo menos 48 horas antes de la prueba, ya que afectan la actividad cerebral.
6. **Objetos metálicos**: Evitar el uso de joyas, horquillas, clips u otros objetos metálicos en el cabello, cara u orejas, ya que pueden interferir con la señal eléctrica registrada.

El procedimiento se realizó dentro de una sala insonora y consistió en que cada sujeto sano repose en una silla y tome una postura de relajación. Se colocó una pantalla de ordenador a una distancia prudente del sujeto y se empleó para mostrar diversas señales visuales. En concreto, cada pista visual notifica al sujeto que imagine la realización de una tarea manual específica. Durante el experimento, cada pista visual notifica al sujeto que imagine la realización de un movimiento o desplazamiento hacia una dirección específica. Las tareas consideradas se pueden agrupar en cinco categorías o clases. Las pistas consideradas fueron: (1) Adelante, (2) Atrás, (3) Derecha, (4) Izquierda y (5) Pare.

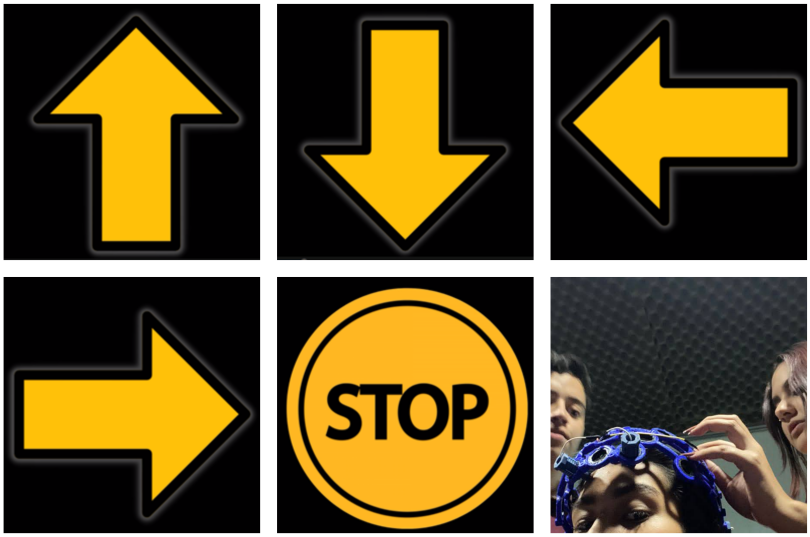

Figura 16

Imagenes de muestra de las señales visuales empleadas por cada tarea.

Fuente: Propia

Durante cada procedimiento se extrajo una serie de señales por un tiempo

aproximado, esto se estableció mediante una fórmula para determinar el número de muestras, teniendo en cuenta que el equipo OpenBCI por defecto admite 250 muestras por segundo y ocho canales se obtiene lo siguiente:

$$Nmuestras = t * 250Hz \tag{45}$$

Donde t es el tiempo deseado para la realización de una tarea, durante el proyecto se estableció un tiempo pertinente de 20 segundos. (Liu et al., 2021), utilizan una tasa de muestreo de 250 Hz. Al utilizar la fórmula, calculan que se establece una muestra de 40.000 señales por tarea (por los 8 canales), siendo 200.000 el número de datos al completar las 5 tareas descritas.

Figura 17

Recolección de señal. Fuente: Propia

3.3 Procesamiento de la señal EEG

De acuerdo con la base de datos recolectada, se diseñará un algoritmo que ofrece información del conjunto de datos que podría llegar a ser más util para la elaboración del algoritmo predictivo de clasificación.

A.) Filtraciòn

Según lo estudiado, en el proyecto se realizó una filtración de orden 4 con frecuencia de 1-30 Hz, con el objetivo de disminuir el ruido en la señal, pero sin eliminar gran cantidad de información que podría ser relevante para su procesamiento.
Teniendo en cuenta lo anterior, estas señales resultantes, serían el nuevo objeto de estudio del cual se determina un conjunto de datos adecuado y que permita mayor facilidad para comenzar a trabajar con la extracción de características.
Para la construcción del filtro Butterworth, se tomaron en consideración diversas funciones del filtro:

Butter bandpass: Esta función calcula los coeficientes del filtro Butterworth, teniendo en cuenta la frecuencia de corte inferior (lowcut), la frecuencia de corte superior (highcut), la frecuencia de muestreo (fs) y el orden del filtro. Los coeficientes del filtro se almacenan en los arrays b y a.

Butter bandpass filter: Utiliza los coeficientes del filtro calculados en la función anterior para aplicar el filtro de paso de banda a los datos EEG de entrada (data). Los datos filtrados se almacenan en filtereddata.

Para la configuración del filtro, se definen los parámetros del filtro de paso de banda: lowcut: Frecuencia de corte inferior del filtro.

highcut: Frecuencia de corte superior del filtro.

fs: Frecuencia de muestreo de la señal EEG en Hz. order: El orden del filtro Butterworth.

Aplicación del filtro de paso de banda: Finalmente, se aplica el filtro de paso de banda a la señal EEG (df) utilizando la función butter-bandpass-filter. La señal EEG filtrada se almacena en filtered-signal. En la figura 18 se representa gráficamente el filtro aplicado en la señal, eliminando parte del ruido.

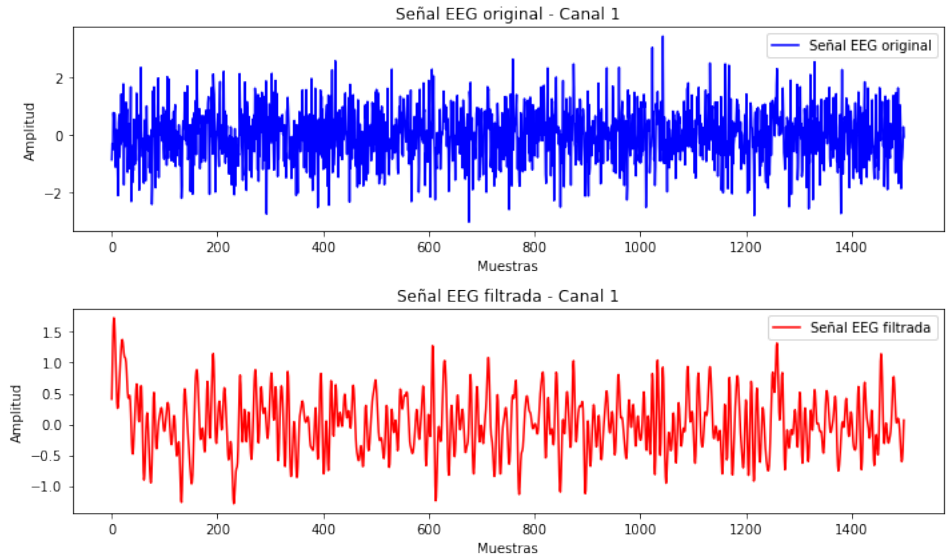

Figura 18

Comparación gráfica de la filtración en el canal 1 del paciente 4. Fuente: Propia

La configuración del filtro se define mediante el siguiente esquema:

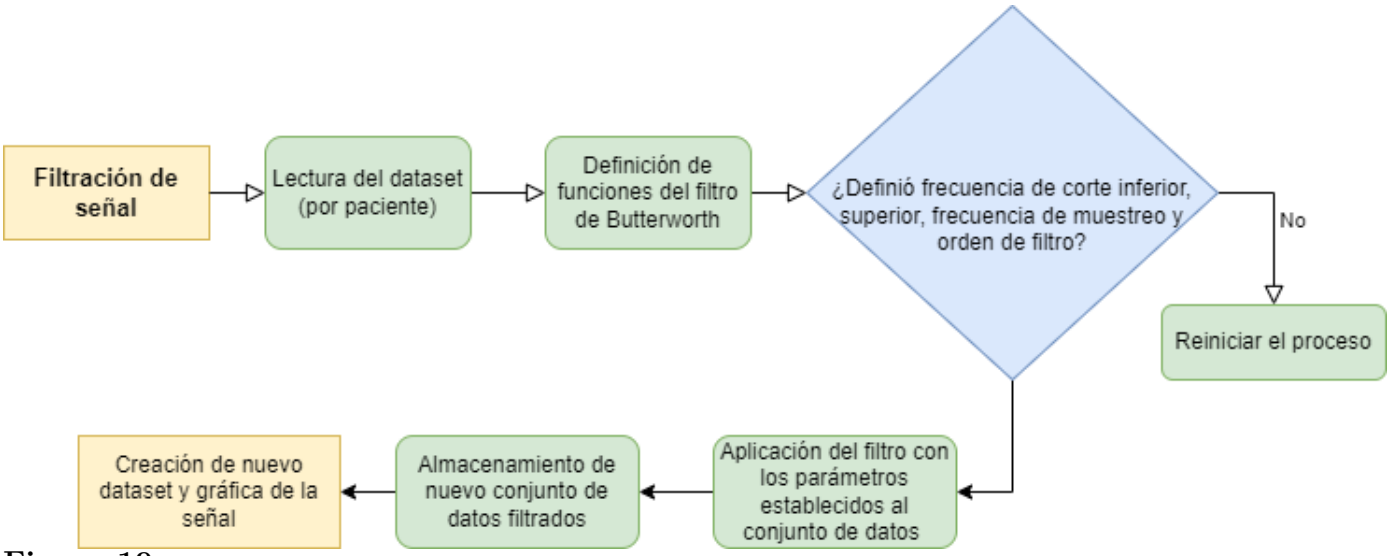

Figura 19

Diagrama de flujo de Filtración de señal. Fuente: Propia

B.) Extracción de características y análisis de componentes principales

Según análisis previos, se realizó una debida selección del mejor paciente mediante un modelo sencillo de entrenamiento con redes convolucionales (CNN) usando las señales EEG puras, es decir, las señales sin haber pasado por el proceso de filtración, con el objetivo de trabajar posteriormente con el método *one to one*, que permitirá conocer la viabilidad del modelo construido por medio del cálculo de las características.

Como resultado anterior, el mejor paciente fué el N° (4), presentando una mayor precisión en comparación con los demás presentando una precisión superior a 0.30. (Fig. 20)

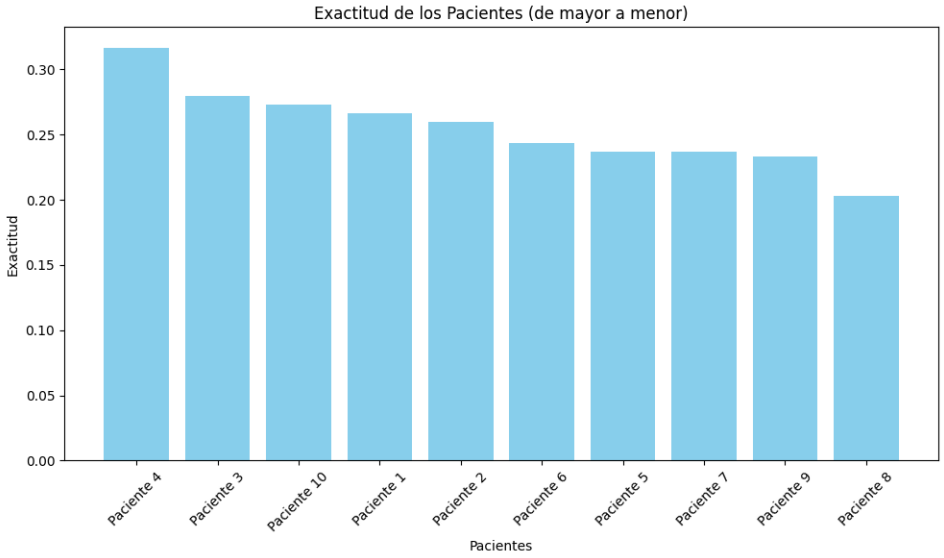

Figura 20

Gráfica de viabilidad del mejor paciente. Fuente: Propia

El proceso de selección del mejor paciente se representa de la siguiente manera:

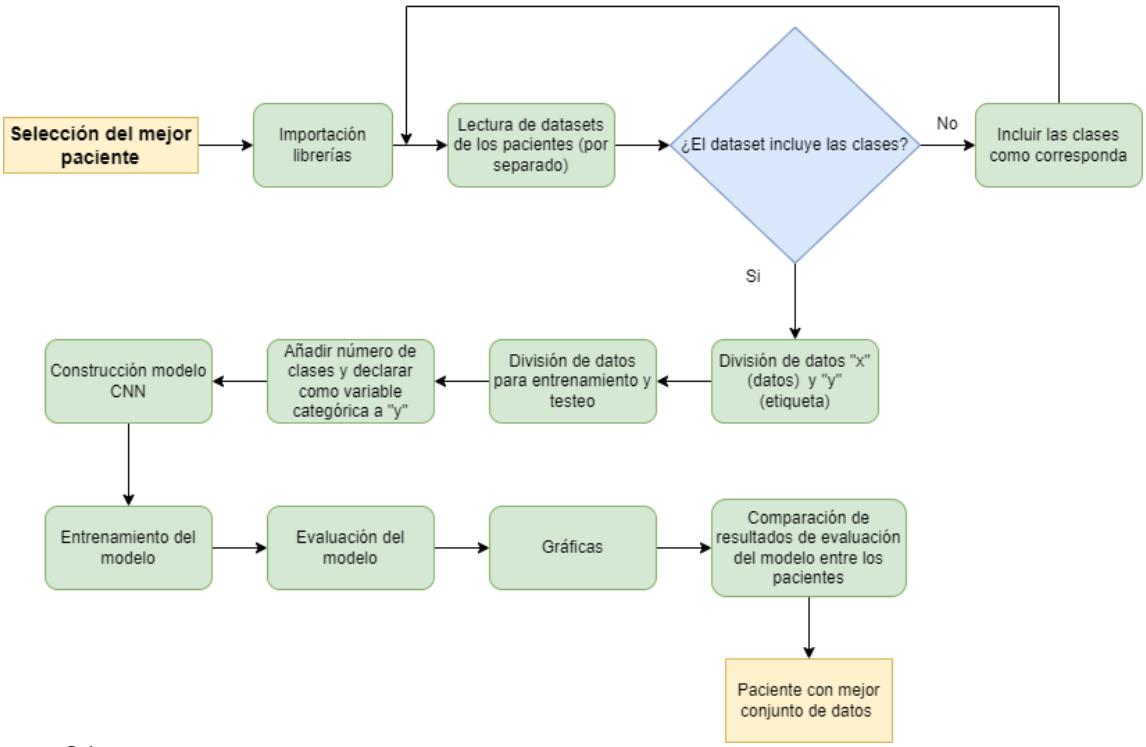

Figura 21

Diagrama de flujo de viabilidad del mejor paciente. Fuente: Propia

Por lo dicho anteriormente, el dataset del paciente N° 4 es el conjunto de datos propicio para la extracción de características con el objetivo de identificar patrones y que el programa sea capaz de aprender de estos datos mediante el cálculo de las características en el dominio del tiempo, segmentación de datos, extracción y su pertinente almacenamiento como se describe a continuación:

- Definición de funciones de extracción de características:

El código comienza con la definición de varias funciones que se utilizan para extraer características de señales. Estas funciones incluyen cálculos como el Valor Absoluto Medio, Longitud de Onda, Cruce por Cero, Raíz Cuadrada Media, Variación Media

de la Amplitud, etc. Cada función toma una señal (serie de datos) como entrada y calcula una característica específica.

- Segmentación de datos:

Las señales EEG son altamente dinámicas y cambian con el tiempo debido a la actividad cerebral. Dividir la señal en segmentos más pequeños (por ejemplo, de 50 unidades de tiempo) con un solapamiento de 25 unidades de tiempo permite analizar patrones temporales en la señal. Esto es útil para identificar eventos, como picos o patrones específicos en la actividad cerebral.

En EEG, hay eventos cerebrales que son de corta duración pero de gran importancia, como las ondas cerebrales asociadas con la aparición de un pico de epilepsia. El solapamiento de 25 unidades de tiempo asegura que estos eventos breves no se pierdan en la segmentación.

Cada segmento puede considerarse como una ventana de tiempo en la que se extraen características, como la potencia en diferentes bandas de frecuencia, la coherencia o la conectividad funcional.

Se define una función llamada segmentar-datos que se utiliza para dividir una serie de datos en segmentos más pequeños con un cierto tamaño y solapamiento. Esto es útil para analizar las señales en fragmentos más manejables.

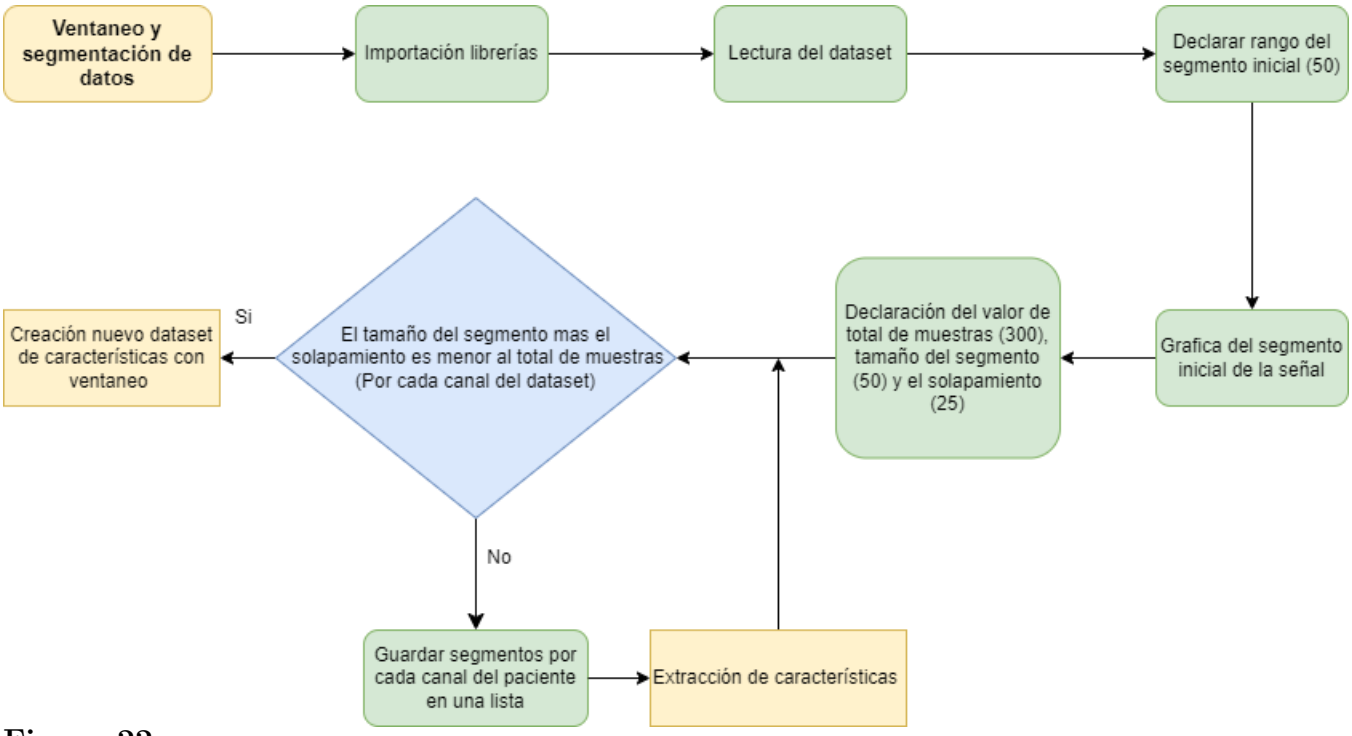

Figura 22

Diagrama de flujo de lectura, segmentación y almacenamiento de los datos. Fuente: Propia

- Lectura de datos:

Se utiliza la biblioteca pandas para leer un archivo CSV (Paciente.csv) en un DataFrame.

- Lista de características:

Se define una lista llamada FEATURES que contiene los nombres de las características que se van a extraer de las señales.

• Extracción de características por canal:

El código realiza un bucle a través de ocho canales y extrae características para cada segmento de cada señal. Las características extraídas incluyen Emav, Ewl, Mav, Wl, Zc, Rms, Aac, Ssc, Dasdv, Ld, Mmav, Mmav2, Wamp, Ssi, Var, Mfl, Hjorst1, Hjorst2, CD, dfa, Hurstt, y pfd.

• Almacenamiento de datos:

Las características extraídas se almacenan en listas separadas (por característica) y se agrupan juntas en una matriz dataM. Luego, se crea un DataFrame de pandas llamado df-features que contiene estas características junto con sus nombres.

• Guardar en un archivo CSV:

Finalmente, el DataFrame df-features se guarda en un archivo CSV (posiblemente especificado por input) para su posterior análisis o uso en otras partes del trabajo. Como resultado, se obtuvo un dataset con las características, siendo el nuevo conjunto de datos pertinente para el análisis de componentes principales y elaborar matrices de correlación.

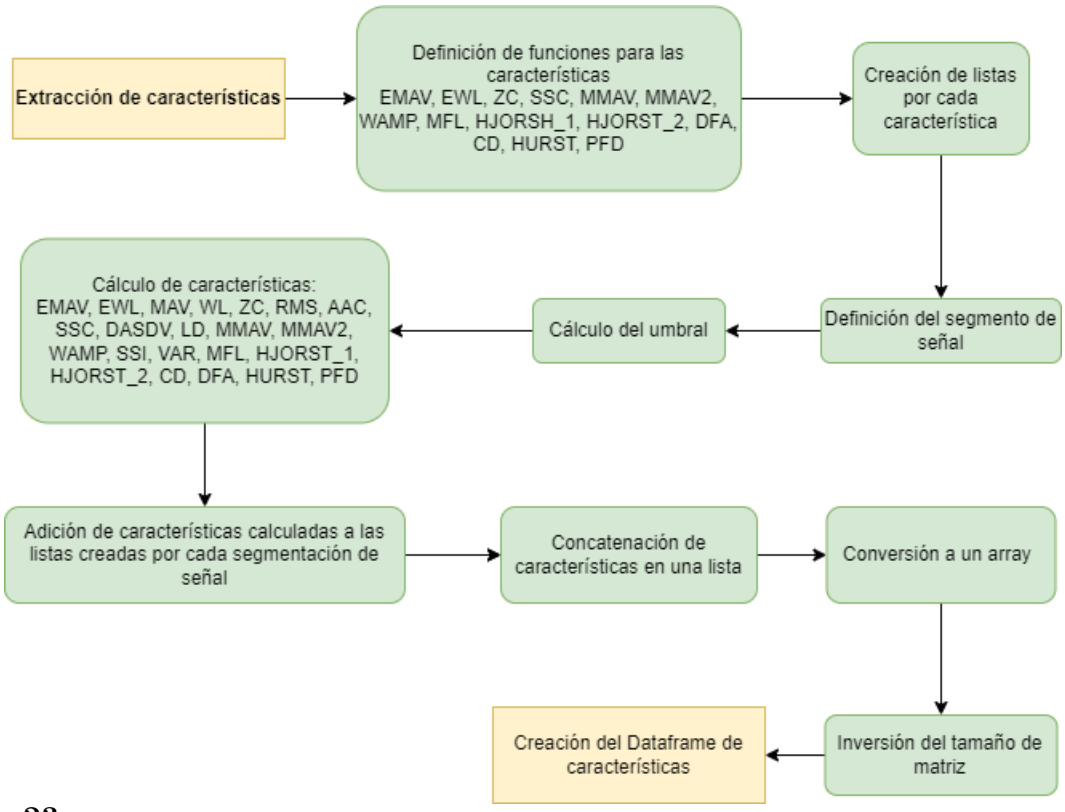

Figura 23

Diagrama de flujo de extracción de características. Fuente: Propia

C.) Análisis de datos

¿Por qué se realizan análisis del PCA y matrices de correlacion y covarianza?

Algunas de las ventajas del análisis propuesto son las siguientes:

1. Reducción de la dimensionalidad: Con 20 características, el espacio de características es relativamente grande, lo que puede llevar a problemas de maldición de la dimensionalidad y un aumento en el tiempo de cálculo. Al seleccionar solo 5 características (la mejor y las 4 peores), se reduce significativamente la

dimensionalidad del problema, lo que puede acelerar el entrenamiento del modelo y reducir el riesgo de sobreajuste.

2. Eliminación de características irrelevantes: No todas las características en un conjunto de datos son igualmente informativas. Al seleccionar la mejor característica, se asegura de que al menos una característica relevante esté presente. Al seleccionar las 4 peores características, se eliminan las características menos relevantes o incluso ruidosas. Esto puede mejorar la capacidad del modelo para generalizar a nuevos datos.

3. Interpretabilidad: En algunos casos, seleccionar la mejor característica y las 4 peores puede hacer que el modelo sea más interpretable. Es más fácil analizar e interpretar un modelo basado en unas pocas características que en un modelo que utiliza todas las características originales.

4. Eficiencia computacional: Al reducir la dimensionalidad del espacio de características, el modelo SVM (u otro modelo) se ejecutará más rápido y requerirá menos recursos computacionales. Esto es especialmente importante en aplicaciones en tiempo real o con grandes conjuntos de datos.

Metología para selección de características

- Normalización de características:

scaler = StandardScaler(): Se crea un objeto StandardScaler, que se utiliza para estandarizar (normalizar) las características. La normalización es importante en PCA para asegurarse de que todas las características tengan la misma escala.
X-normalized = scaler.fit-transform(X): Se aplica la normalización a los datos originales X, lo que significa que cada característica tendrá una media de 0 y una desviación estándar de 1.

- Calculo de la Matriz de Correlación:

En esta línea de código, se calcula la matriz de correlación. Para hacerlo, primero se elimina la columna de etiquetas del DataFrame (df) utilizando df.drop(columna-etiquetas, axis=1). Luego, se aplica la función .corr() para calcular la correlación entre las variables restantes en el DataFrame. La matriz de correlación es una tabla que muestra cómo las variables se relacionan entre sí, lo que es útil para identificar patrones y asociaciones en los datos.
Visualizar la Matriz de Correlación:
En esta parte, se visualiza la matriz de correlación utilizando una gráfica de calor (heatmap) para resaltar las relaciones entre las variables.

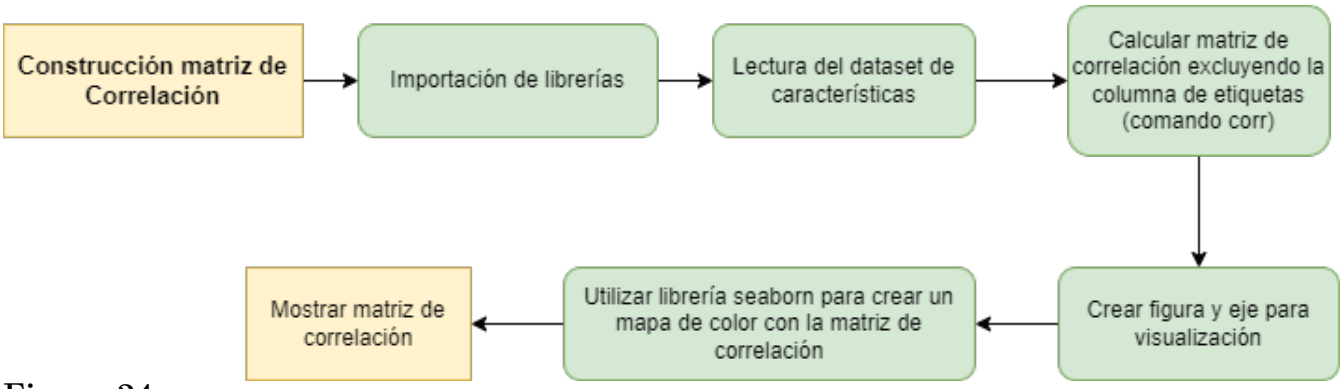

Figura 24

Diagrama de flujo de de construcción de la matriz de correlación. Fuente: Propia

- PCA (Análisis de Componentes Principales):

pca = PCA(): Se crea un objeto PCA, que se utilizará para realizar el análisis de componentes principales en los datos normalizados.

pca.fit(X-normalized): Se ajusta el modelo PCA a los datos normalizados. Esto calcula los componentes principales y sus respectivas varianzas explicadas.

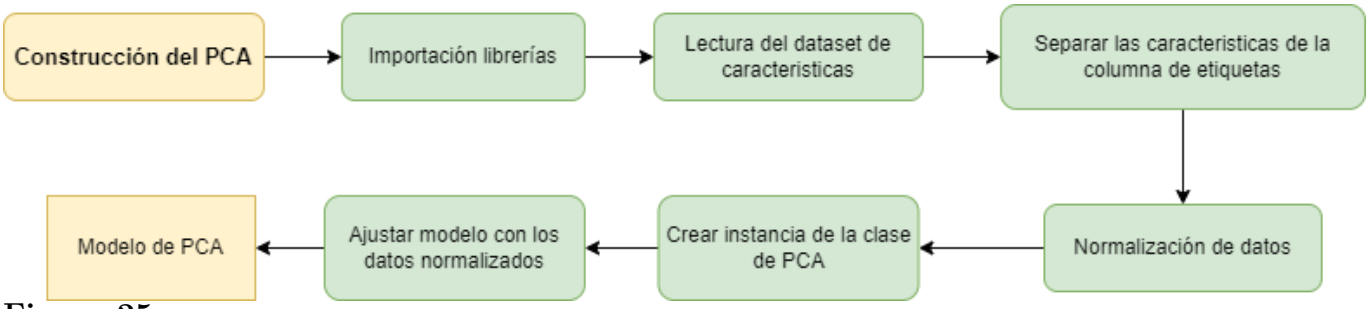

Figura 25

Diagrama de flujo de construcción del PCA. Fuente: Propia

• Cálculo de la Matriz de Covarianza:

cov-matrix = np.cov(X-normalized, rowvar=False): Se calcula la matriz de covarianza a partir de los datos normalizados. La matriz de covarianza muestra las relaciones lineales entre las características y es fundamental para PCA, ya que PCA busca los ejes de máxima varianza en los datos.

Visualización de la matriz de covarianza:

plt.figure(figsize=(12, 12)): Se crea una figura de matplotlib para la visualización.

sns.heatmap(cov-matrix, annot=True, fmt=".2f", cmap=çoolwarm", square=True): Se utiliza Seaborn para crear un mapa de calor de la matriz de covarianza. Esto permite visualizar las relaciones entre las características, donde los colores representan la magnitud y la dirección de las covarianzas. plt.title("Matriz de Covarianza"): Se establece un título para la figura.

plt.yticks(range(len(feature-names)), feature-names, rotation=0): Se agregan etiquetas a lo largo del eje y, que representan las características, y se establece su orientación.

n-componentes = 22: Se define el número de componentes principales a mostrar en el eje x.

component-labels = [f'PC(i+1)' for i in range(n-componentes)]: Se crean etiquetas para los componentes principales.

plt.xticks(range(n-componentes), component-labels, rotation=45): Se establecen etiquetas en el eje x, que representan los componentes principales, y se rota su orientación.

plt.xlabel(Componentes Principales): Se agrega una etiqueta al eje x.

plt.ylabel(Características): Se agrega una etiqueta al eje y.

plt.show(): Finalmente, se muestra la visualización de la matriz de covarianza.

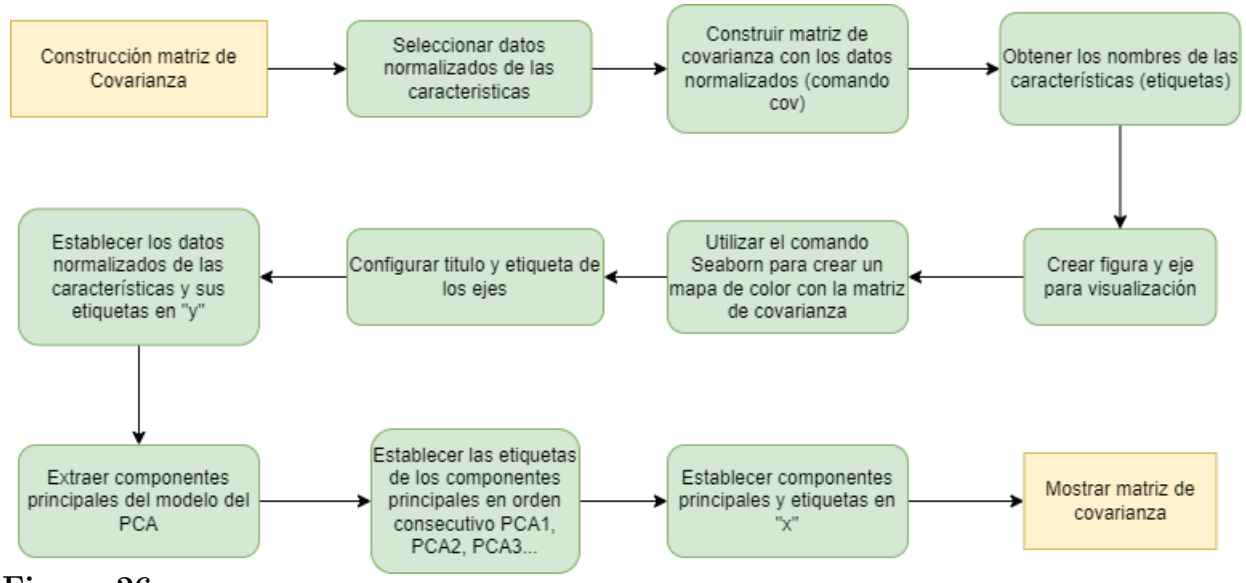

Figura 26

Diagrama de flujo de construcción de la matriz de covarianza. Fuente: Propia

- Selección de características:

Para saber cuál es la mejor y las 4 peores características se observan las componentes principales y sus correspondientes cargas factoriales. Las cargas factoriales indican cómo cada característica contribuye a cada componente principal. Características con cargas factoriales altas (positivas o negativas) en un componente específico tienen una mayor influencia en ese componente y, por lo tanto, son más significativas en términos de varianza explicada.

El método propuesto al seleccionar la mejor y las 4 peores características trae grandes ventajas tales como:

1. Balance entre información relevante y redundancia: La idea detrás de la selección de la mejor característica y las peores es encontrar un equilibrio entre la

característica más informativa (la mejor) y las características que aportan información menos valiosa o incluso redundante (las peores). Esto puede ayudar a reducir el riesgo de sobreajuste, ya que las características redundantes pueden introducir ruido en el modelo.

2. Balance entre subrepresentación y sobrerepresentación: Al elegir la mejor característica y las peores, se busca evitar subrepresentar el espacio de características (usando solo la mejor característica) y, al mismo tiempo, evitar sobrerepresentarlo (usando solo las 5 mejores). Esto se hace en un esfuerzo por encontrar un equilibrio que permita al modelo generalizar mejor sin introducir complejidad innecesaria.

3. Conservación de información relevante: Incluso las características que no son las mejores pueden contener información útil o contribuir de alguna manera al rendimiento del modelo. Al seleccionar la mejor y algunas de las peores características, se conserva una parte de esa información, lo que puede ser beneficioso para el rendimiento general del modelo.

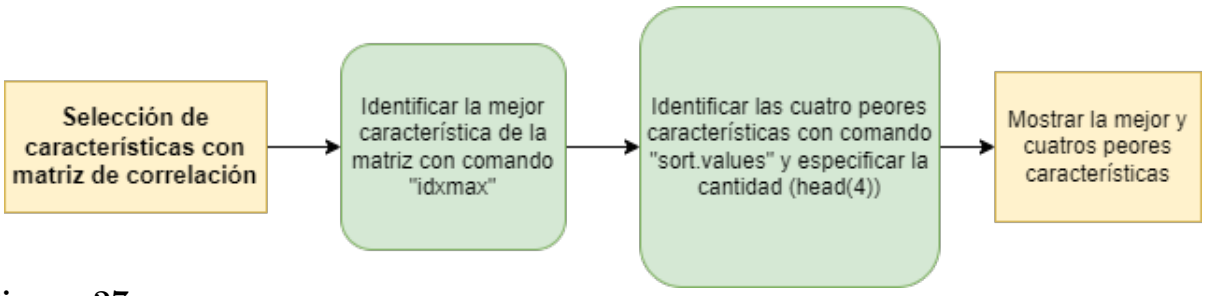

Figura 27

Diagrama de flujo de selección de características con matriz de correlación. Fuente: Propia

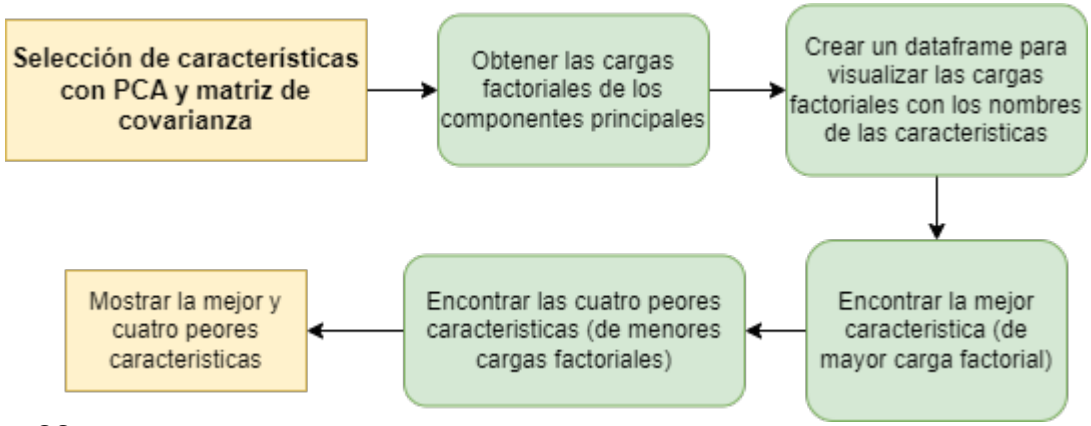

Figura 28

Diagrama de flujo de selección de características con PCA. Fuente: Propia

3.4 Clasificación

Construcción:

- Para la construcción del algoritmo clasificador se deben importar determinadas bibliotecas necesarias para el análisis de datos, la implementación del modelo SVM y la visualización de resultados. El diagrama de flujo que resume el proceso se encuentra en la figura 29.
- Se selecciona el conjunto de datos (datos de las características) desde un archivo CSV a un DataFrame de pandas y se muestra una vista previa de las primeras filas del DataFrame. Se realiza la respectiva partición de los datos en características (x) y etiquetas (y) utilizando la columna identificada anteriormente como la columna de etiquetas.
- Se determina el número de clases en el conjunto de datos y se crea una lista de nombres de clases.)
- Se verifica el desequilibrio de las etiquetas antes de aplicar técnicas de balanceo. Se calcula el desequilibrio actual y se establece un umbral para decidir si se aplicará la técnica de sobremuestreo SMOTE. Se aplica SMOTE para equilibrar las clases si el desequilibrio supera el umbral.
- Se crea una copia de la matriz de características original antes de la normalización. El siguiente bucle itera sobre todas las columnas de características y normaliza solo aquellas que tienen valores fuera del rango [-1, 1]. La normalización se realiza utilizando el escalador MinMaxScaler Se verifica si la normalización se realizó correctamente comparando la matriz original y la matriz normalizada.
- Los datos se dividen en conjuntos de entrenamiento, prueba y validación utilizando

train-test-split.

- Se imprime el tamaño de los conjuntos de datos de entrenamiento, prueba y validación.

Los conjuntos de datos se convierten en matrices NumPy para su uso en el modelo SVM.

- Se crea un clasificador SVM con parámetros específicos y se entrena con los datos de entrenamiento.

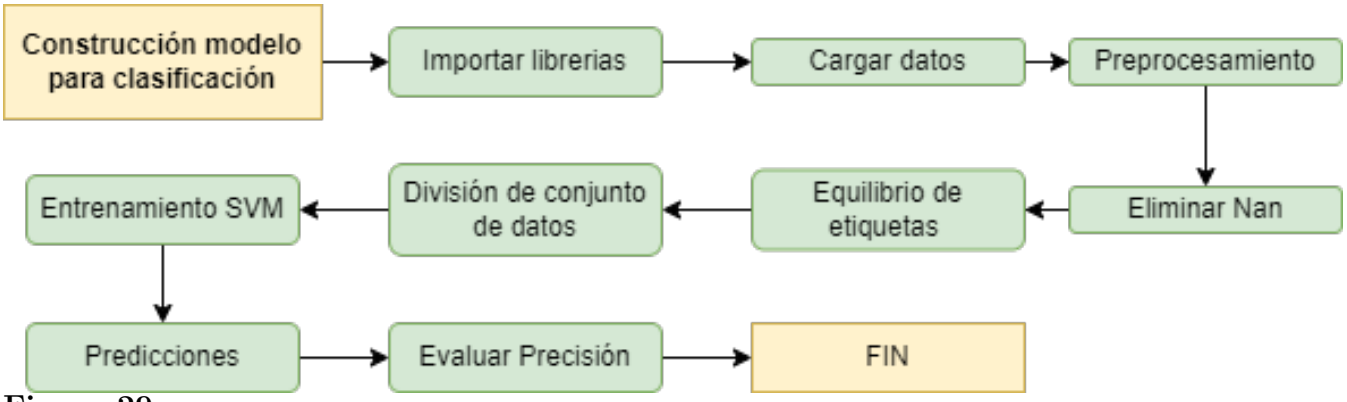

Figura 29

Diagrama de flujo de construcción del Clasificador SVM. Fuente: Propia

Hiperparametros:

Antes de sintonizar los parámetros, se hace un balance de clase y se normaliza por columna si es necesario. Igual que en el archivo de SVM se maneja el mismo código de normalización y balanceo. También, se divide datos en entrenamiento, testeo y validación.

- Sección 1: Núcleo Lineal

En esta sección, se busca ajustar un modelo SVM con núcleo lineal y encontrar los mejores hiperparámetros. Los pasos son los siguientes: Se define un diccionario param-grid que especifica los valores posibles para el hiperparámetro C. Se crea un objeto GridSearchCV llamado grid que se encarga de la búsqueda exhaustiva de hiperparámetros. El modelo SVM se ajusta con un núcleo lineal. Los parámetros de GridSearchCV incluyen el estimador, los hiperparámetros a buscar, la métrica de evaluación (precisión en este caso), el número de trabajadores (n-jobs) para paralelizar el proceso, el número de divisiones en la validación cruzada (cv), y otros. El modelo se ajusta al conjunto de entrenamiento (x-train e y-train) utilizando la función fit. Los resultados de la búsqueda de hiperparámetros se almacenan en un DataFrame llamado resultados. Se filtran y muestran los mejores resultados.

- Sección 2: Núcleo rbf (Radial Basis Function)

En esta sección, el proceso se repite para un modelo SVM con núcleo rbf. Se buscan los mejores valores de los hiperparámetros C y gamma.

- Sección 3: Núcleo Polinómico Esta sección se centra en el núcleo polinómico del SVM. Se buscan los mejores valores de los hiperparámetros C, gamma, degree y coef0.

- Sección 4: Núcleo Sigmoide

Finalmente, en esta sección, se realiza una búsqueda de hiperparámetros para un

modelo SVM con núcleo sigmoide. Los hiperparámetros que se buscan son C, gamma y coef0.

En todas las secciones, se sigue un proceso similar:

Se define un diccionario param-grid con los valores posibles para los hiperparámetros del núcleo correspondiente. Se crea un objeto GridSearchCV para ajustar el modelo SVM con el núcleo respectivo y buscar los mejores hiperparámetros. Los resultados se almacenan en DataFrames y se muestran los mejores resultados. Al final de cada sección, se imprimen los mejores hiperparámetros encontrados y se muestra el aspecto del modelo SVM ajustado después de la búsqueda de hiperparámetros.

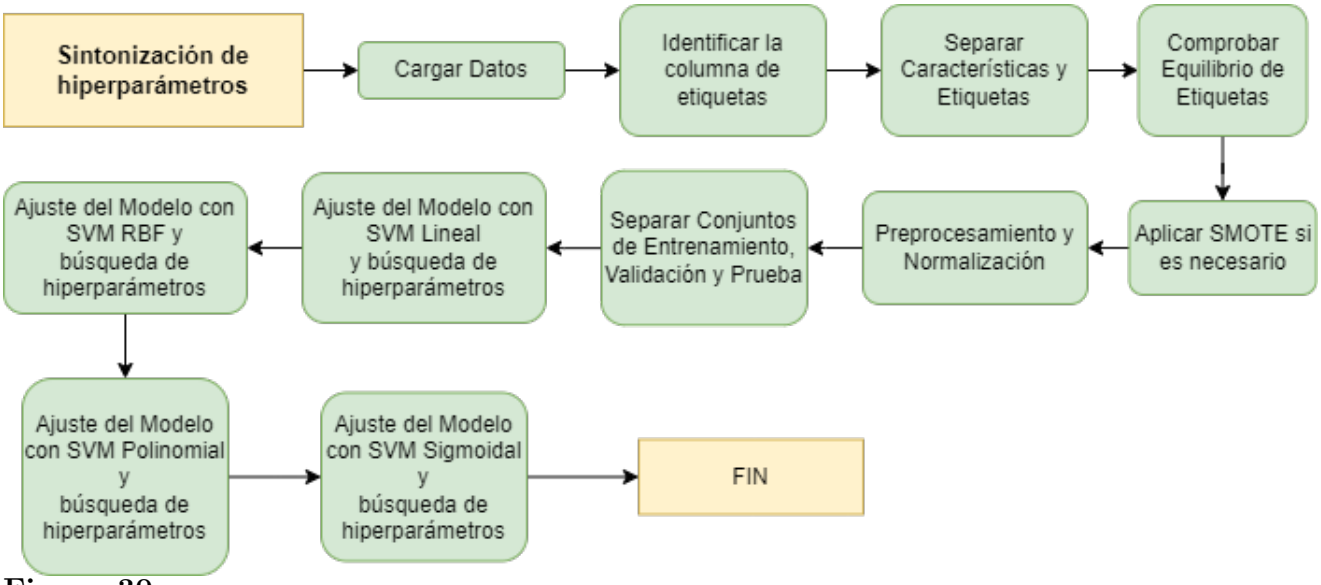

Figura 30

Diagrama de flujo de sintonización de hiperparametros. Fuente: Propia

El Cuadro 5 detalla los rangos específicos considerados para cada hiperparámetro en la exploración de los diferentes tipos de kernel en el clasificador SVM. Estos rangos representan los valores posibles que fueron investigados para encontrar la

combinación óptima de hiperparámetros que maximizara la precisión del modelo en la clasificación de datos.

1. Kernel Polinomial (Poly): Se evaluaron valores de regularización (C) en el rango de 0.1 a 1000, grados (Degree) del polinomio de 1 a 5, gamma (Gamma) y coef0 en los rangos de 1 a 4.
2. Kernel Sigmoid (Sigmoid): Se investigaron valores de C en el rango de 0.1 a 1000, grados y gamma en el rango de 1 a 4, y coef0 en el rango de 1 a 4.
3. Kernel RBF (Radial Basis Function): Se exploraron valores de C en el rango de 0.1 a 1000, gamma en el rango de 1 a 4, y no se consideró el coef0 para esta función de kernel.
4. Kernel Lineal (Linear): Los valores de C considerados estuvieron en el rango de 0.1 a 1000, ya que el kernel lineal no utiliza parámetros adicionales como gamma o coef0.

Kernel	**Regularización**	**Gamma**	**Coef0**	**Degree**
Poly	0.1,1,10,100,1000	1,2,3,4	1,2,3,4	1,2,3,4,5
Sigmoid	0.1,1,10,100,1000	1,2,3,4	1,2,3,4	
RBF	0.1,1,10,100,1000	1,2,3,4		
Linear	0.1,1,10,100,1000			

Cuadro 5

Rango de hiper parámetro para el clasificador.

Fuente: Propia

Estos rangos fueron seleccionados para abarcar una amplia gama de posibilidades y permitir una búsqueda exhaustiva de los hiperparámetros que resultaran en el mejor rendimiento del clasificador SVM para cada tipo de kernel.

El Cuadro 6 detalla los resultados obtenidos después de la búsqueda de hiperparámetros. Este muestra los valores seleccionados para los hiperparámetros y la precisión correspondiente alcanzada por el modelo SVM para cada tipo de kernel.

Kernel	**Regularización**	**Gamma**	**Coef0**	**Degree**	**Accuracy**
Poly	1000	4	4	4	0.62
Sigmoid	1000	4	4		0.21
RBF	1000	4			0.79
Linear	1				0.35

Cuadro 6

Hiperparametros seleccionados

Fuente: Propia

Sección 4

4. RESULTADOS

Al obtener las características previamente estudiadas, se elaboró la matriz de correlación, como se muestra en la Figura 31. Por otro lado, se representa un análisis de componentes principales y se representaron mediante una matriz de covarianza como se muestra en la Figura 32. Seguidamente, se seleccionaron las cuatro características con menor correlación y una característica con la mayor correlación, teniendo en cuenta las relaciones más significativas de los datos y el porcentaje de varianza explicada acumulada que se lograban analizar del PCA y la matriz mencionada.

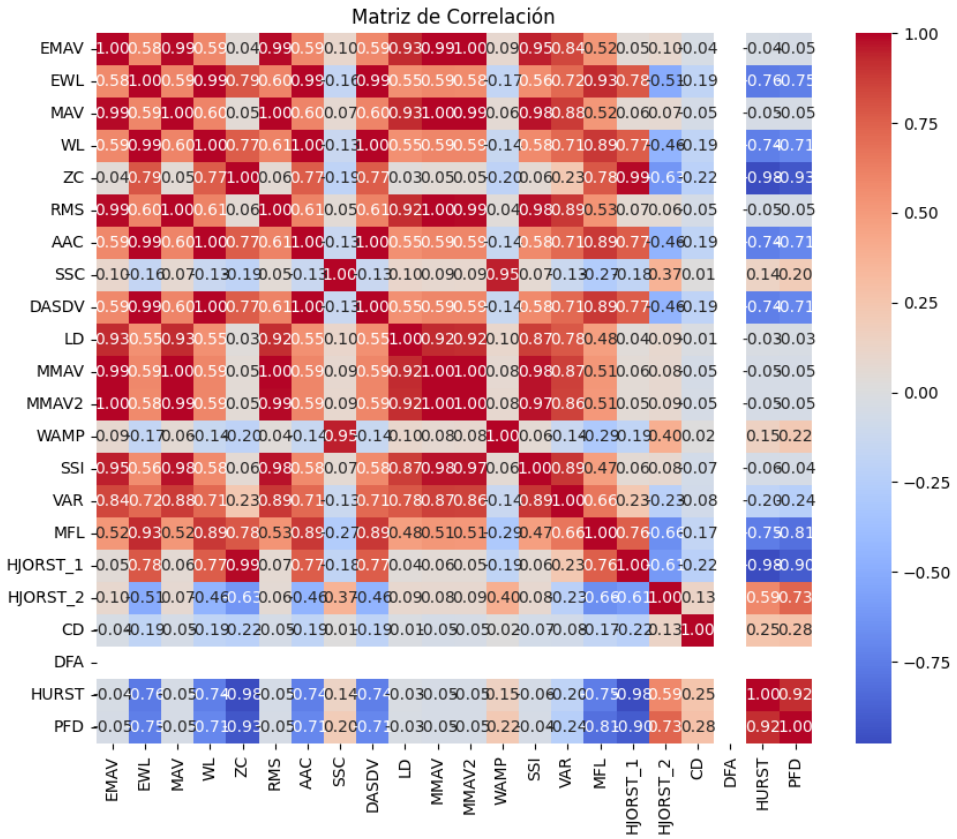

Figura 31

Matriz de correlación. Fuente: Propia

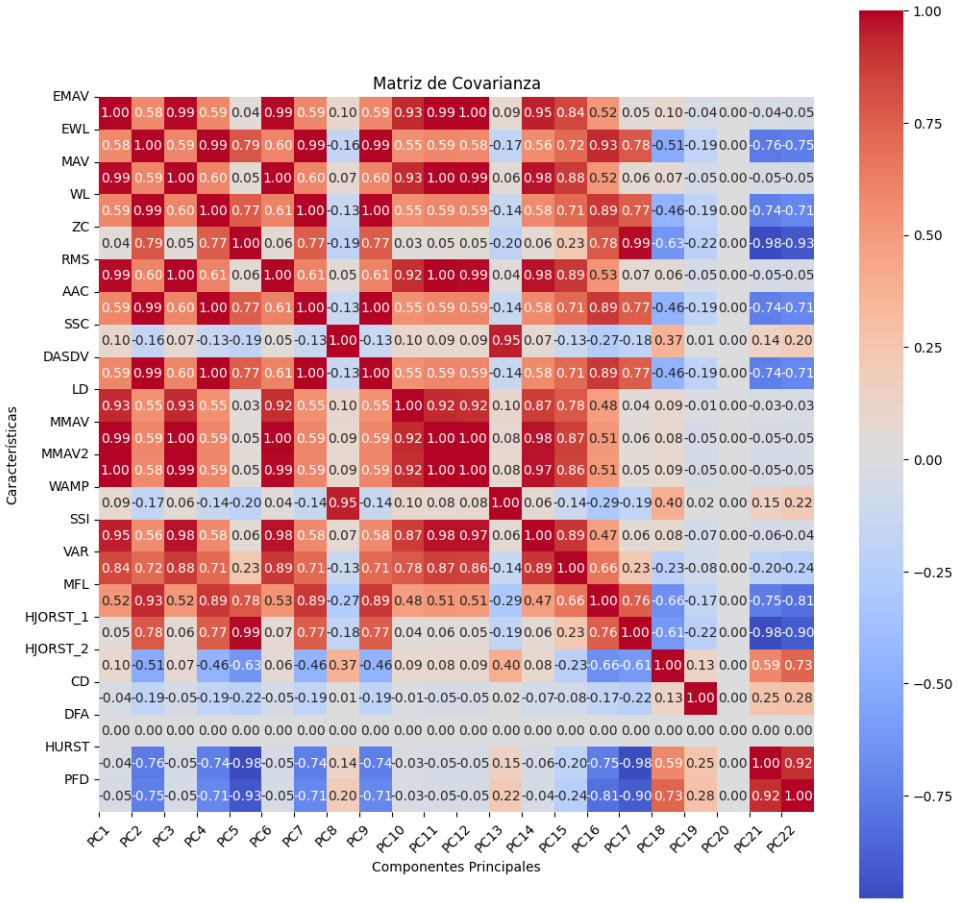

Figura 32

Matriz de covarianza con datos del PCA. Fuente: Propia

¿Por qué son diferentes las características en la correlación y en el PCA?

Las características más importantes pueden variar dependiendo de los datos y el método utilizado para la selección, y el enfoque de PCA se basa en la variabilidad de los datos en lugar de la correlación. Es válido utilizar esas características identificadas como las más importantes. PCA, por otro lado, selecciona automáticamente las componentes principales basadas en la varianza de los datos y no necesariamente

coincide con las características específicas.

En cuanto al análisis de la matriz de correlación, se puede determinar una correlación directa e inversa (Se diferencia visualmente por los colores y el signo), por lo tanto, es importante resaltar que independientemente de que sea una relación lineal negativa, puede sostener un gran porcentaje de correlación y por ello, eso se tuvo en cuenta durante la selección de características.

En otras palabras, se realizó una reducción de variables utilizando este análisis y en el caso de la característica con mayor correlación fue basada en tener el porcentaje más cercano al 100 % de varianza y las cuatro peores con valores más cercanos al 0 %. Por lo tanto, las características seleccionadas se convierten en los datos pertinentes para ser usados en el modelo de clasificación.

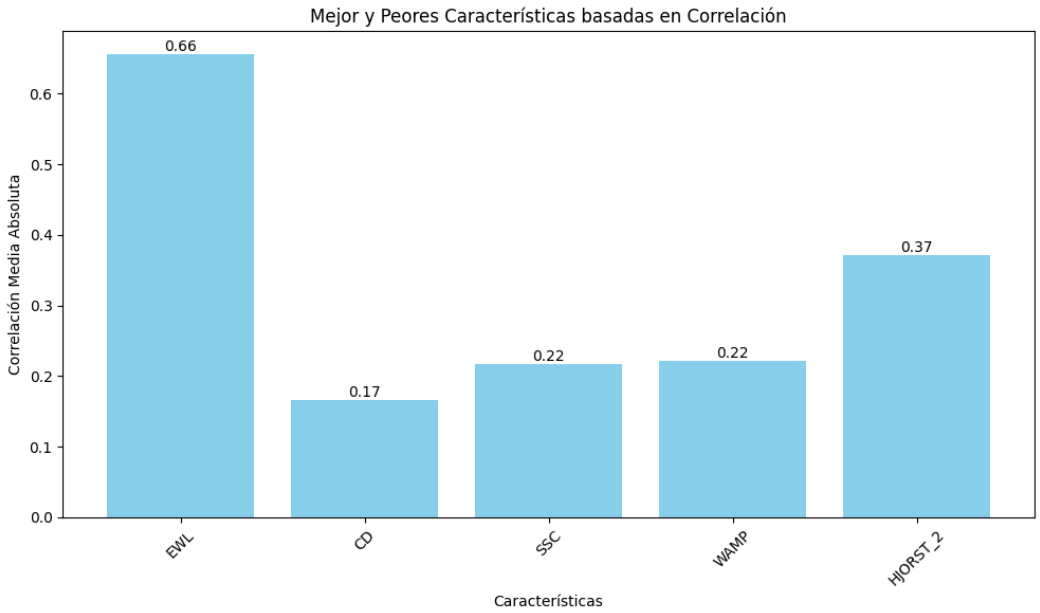

Figura 33

Mejor características y 4 peores características. Fuente: Propia

Se realizó la clasificación mediante un modelo de máquina de vector de soporte (SVM) con método one to one para el paciente N°4, obteniendo una precisión del 0.79. Se obtuvo una matriz de confusión tanto para testeo y validación como la mostrada en la Figura 34 y 35 respectivamente.

En la matriz de confusión los números en la diagonal principal de la matriz representan las predicciones correctas del modelo.

Los números en las celdas fuera de la diagonal principal indican errores en las predicciones del modelo de los cuales surgen los falsos positivos y los falsos negativos.

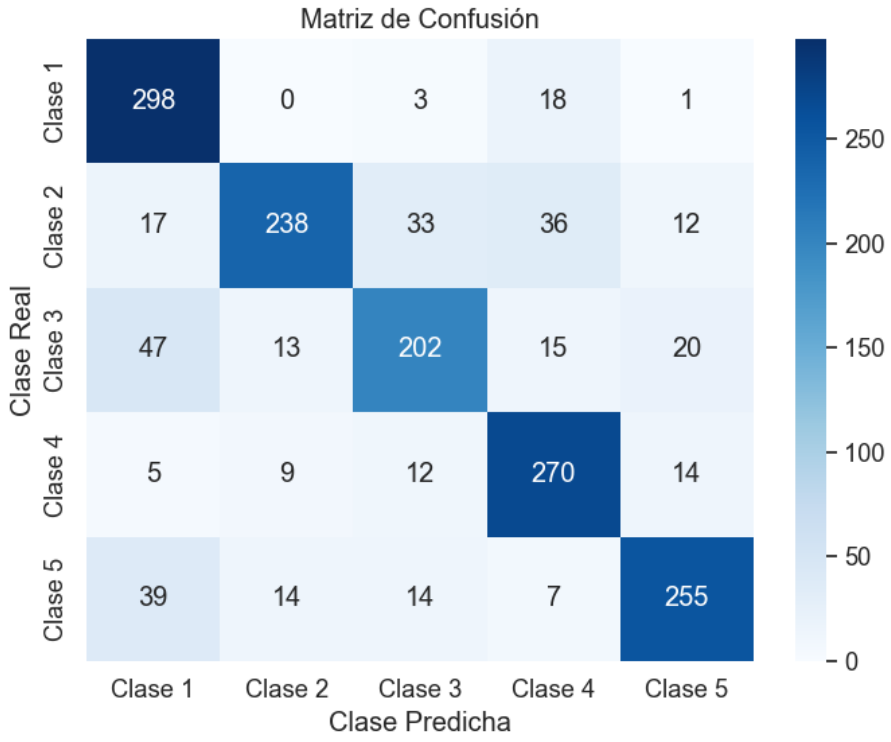

Figura 34

Matriz de confusión con datos de testeo. Fuente: Propia

Para interpretar la matriz de confusión con datos de testeo (Figura 34):

1. La clase 1 tiene 298 predicciones correctas (clasificaciones verdaderas).
2. La clase 2 tiene 238 predicciones correctas.
3. La clase 3 tiene 202 predicciones correctas.
4. La clase 4 tiene 270 predicciones correctas.
5. La clase 5 tiene 255 predicciones correctas.

Además, se pueden observar los errores de clasificación:

1. Para la clase 1, hay 22 instancias mal clasificadas: 3 se clasificaron como clase 3, 18 como clase 4, y 1 como clase 5.
2. Para la clase 2, hay 98 instancias mal clasificadas: 17 se clasificaron como clase 1, 33 como clase 3, 36 como clase 4, y 12 como clase 5.
3. Para la clase 3, hay 95 instancias mal clasificadas: 47 se clasificaron como clase 1, 13 como clase 2, 15 como clase 4, y 20 como clase 5.
4. Para la clase 4, hay 50 instancias mal clasificadas: 5 se clasificaron como clase 1, 9 como clase 2, 12 como clase 3, y 14 como clase 5.
5. Para la clase 5, hay 74 instancias mal clasificadas: 39 se clasificaron como clase 1, 14 como clase 2, 14 como clase 3, y 7 como clase 4.

Para interpretar la matriz de confusión con datos de Validación (Figura 35):

1. La clase 1 tiene 298 predicciones correctas.
2. La clase 2 tiene 224 predicciones correctas.

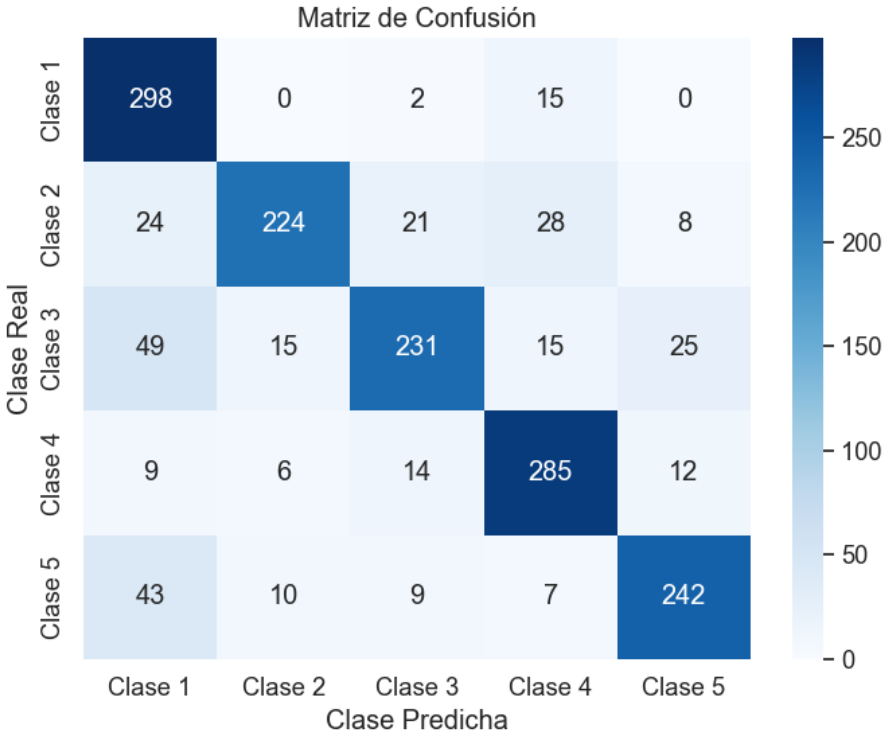

Figura 35

Matriz de confusión con datos de validación. Fuente: Propia

3. La clase 3 tiene 231 predicciones correctas.
4. La clase 4 tiene 285 predicciones correctas.
5. La clase 5 tiene 242 predicciones correctas.

A continuación, se detallan los errores de clasificación:

1. Para la clase 1, hay 17 instancias mal clasificadas: 2 se clasificaron como clase 3 y 15 como clase 4.
2. Para la clase 2, hay 81 instancias mal clasificadas: 24 se clasificaron como clase

1, 21 como clase 3, 28 como clase 4, y 8 como clase 5.

3. Para la clase 3, hay 104 instancias mal clasificadas: 49 se clasificaron como clase 1, 15 como clase 2, 15 como clase 4, y 25 como clase 5.
4. Para la clase 4, hay 41 instancias mal clasificadas: 9 se clasificaron como clase 1, 6 como clase 2, 14 como clase 3, y 12 como clase 5.
5. Para la clase 5, hay 69 instancias mal clasificadas: 43 se clasificaron como clase 1, 10 como clase 2, 9 como clase 3, y 7 como clase 4.

La curva ROC es una forma útil de evaluar la precisión de las predicciones de modelo al trazar la sensibilidad frente a la especificidad de una prueba de clasificación, por lo tanto es viable representar gráficamente el modelo con el conjunto de testeo en la Figura 36 y validación en la Figura 37.

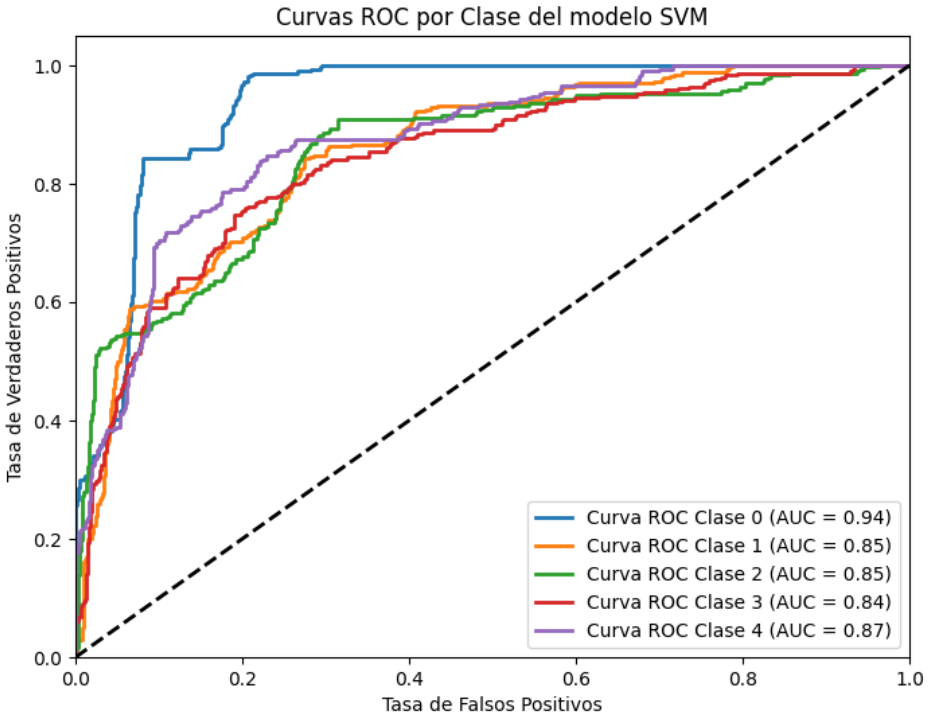

Figura 36

Curva ROC con datos de testeo. Fuente: Propia

Estos valores de AUC muestran cómo el modelo SVM se desempeña en la clasificación de cada clase. interpretación general:

1. Para la Clase 0: Un AUC de 0.94 indica que el modelo tiene una excelente capacidad para distinguir entre la Clase 0 y otras clases. Es un rendimiento muy bueno, lo que sugiere que el modelo SVM es muy preciso al clasificar esta clase en particular.

2. Clases 1, 2, 3 y 4: Los valores de AUC para estas clases son un poco más bajos

en comparación con la Clase 0, pero siguen siendo sólidos. Un AUC alrededor de 0.85-0.87 generalmente se considera bastante bueno y sugiere que el modelo SVM tiene una buena capacidad para distinguir estas clases.

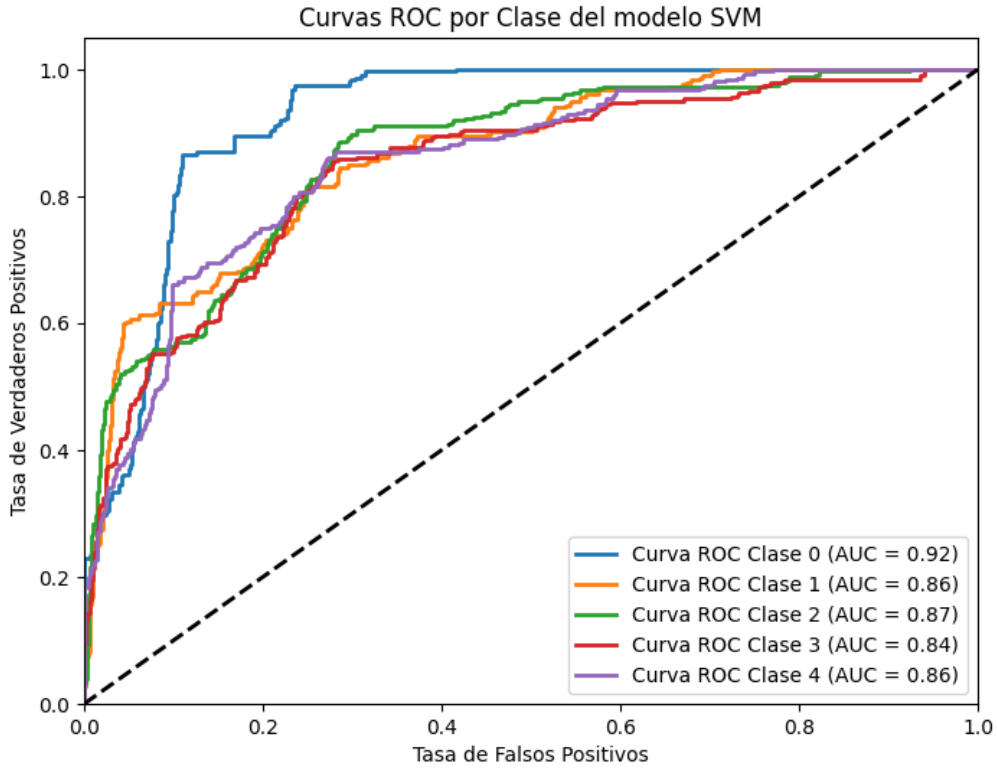

Figura 37

Curva ROC con datos de validación. Fuente: Propia

La evaluación del modelo SVM en los datos de validación muestra resultados similares a los de los datos de prueba o testeo, lo que es positivo en términos de consistencia en el rendimiento del modelo entre conjuntos de datos diferentes.

1. Clase 0: El AUC de 0.92 indica que el modelo sigue teniendo una excelente capacidad para distinguir la Clase 0 en los datos de validación, lo que sugiere que mantiene un rendimiento muy sólido.
2. Clases 1, 2, 3 y 4: Los valores de AUC son ligeramente más bajos que para la Clase 0, pero siguen siendo buenos. Los valores entre 0.84 y 0.87 muestran que el modelo sigue siendo capaz de distinguir estas clases en los datos de validación.

4.1 Métricas

La Tabla 7 presenta un resumen de las métricas esenciales del sistema, lo que ofrece una evaluación general del desempeño del modelo en la tarea de clasificación, tanto para el conjunto de datos para el testeo como para la validación.

Métricas	**Testeo**	**Validación**
Exactitud (Accuracy)	0.79	0.80
Precisión	0.80	0.81
Recuperación(Recall)	0.79	0.80
F1-score	0.79	0.80

Cuadro 7

Métricas en testeo y validación. Fuente: Propia

Testeo	Precision	Recall	F1-score
Clase 1	0.73	0.93	0.82
Clase 2	0.87	0.71	0.78
Clase 3	0.77	0.68	0.72
Clase 4	0.78	0.87	0.82
Clase 5	0.84	0.78	0.81
Validación			
Clase 1	0.70	0.95	0.81
Clase 2	0.88	0.73	0.80
Clase 3	0.83	0.69	0.75
Clase 4	0.81	0.87	0.84
Clase 5	0.84	0.78	0.81

Cuadro 8

Métricas en testeo y validación por clases. Fuente: Propia

1. **Exactitud (Accuracy):** Es la medida general de la precisión del modelo, calculada como la proporción de predicciones correctas sobre el total de predicciones realizadas. Para el conjunto de testeo y validación, el modelo alcanza una exactitud del 79 % y 80 %, respectivamente. Esto significa que en el conjunto de validación, el modelo tiene un rendimiento ligeramente mejor que en el conjunto de testeo.

2. **Precisión:** Indica la proporción de instancias identificadas como positivas que son realmente positivas. Una alta precisión indica que el modelo hace pocas predicciones falsas positivas. En las métricas detalladas por clase, se observa que la precisión varía para cada clase en ambos conjuntos de datos. Para la Clase 2, la precisión es del 87 % en testeo y del 88 % en validación, lo que indica que el modelo tiene una buena capacidad para identificar correctamente los ejemplos de esa clase en ambos conjuntos.

3. **Recuperación (Recall):** Mide la proporción de instancias positivas que fueron correctamente identificadas por el modelo. Una alta tasa de recuperación indica que el modelo puede identificar la mayoría de las instancias positivas. Similar a la precisión, esta métrica varía según la clase. Para la Clase 1, el modelo logra una recuperación del 93 % en testeo y del 95 % en validación, lo que muestra que es efectivo al identificar la mayoría de las instancias de esta clase en ambos conjuntos.

4. **Puntuación F1 (F1-score):** Es la media armónica entre precisión y recuperación. Proporciona un equilibrio entre ambas métricas. Al igual que las métricas anteriores, el F1-score varía para cada clase y entre los conjuntos de

testeo y validación. La Clase 4 tiene un F1-score del 82 % en testeo y 84 % en validación, indicando un buen equilibrio entre precisión y recuperación para esta clase en ambos conjuntos.

La siguiente tabla muestra los niveles de precisión para cada paciente evaluado con el sistema sujeto especifico:

ID	Accuracy	Precisión	Recall	F1-score
Paciente 1	0.56	0.54	0.54	0.54
Paciente 2	0.61	0.60	0.59	0.58
Paciente 3	0.84	0.83	0.83	0.83
Paciente 4	0.79	0.80	0.79	0.79
Paciente 5	0.46	0.47	0.46	0.46
Paciente 6	0.38	0.38	0.38	0.37
Paciente 7	0.22	0.22	0.22	0.22
Paciente 8	0.26	0.25	0.26	0.26
Paciente 9	0.64	0.65	0.64	0.64
Promedio	0.52	0.52	0.52	0.52

Cuadro 9

Evaluaciones Sujetos específicos.

Fuente: Propia

Los valores presentados representan la precisión obtenida al aplicar un modelo de Máquinas de Vectores de Soporte (SVM) a conjuntos de datos individuales de cada paciente en una evaluación sujeto especifico. Los resultados indican la capacidad del

modelo para clasificar correctamente los datos de cada paciente. Se observa una variabilidad considerable en la precisión entre los pacientes evaluados: algunos muestran una alta precisión, como el Paciente 3 y el Paciente 4 con un 84 % y 79 % respectivamente, lo que indica una buena capacidad del modelo para clasificar la mayoría de sus datos. Por otro lado, hay pacientes con menor precisión, como el Paciente 7 y el Paciente 8 con 22 % y 26 % respectivamente, lo que sugiere dificultades significativas del modelo en clasificar correctamente la gran mayoría de los datos de estos pacientes. Estas diferencias pueden deberse a la complejidad de los datos individuales o a la dificultad para distinguir patrones específicos en cada caso. Estos resultados resaltan la importancia de analizar y mejorar la capacidad del modelo para clasificar eficazmente los datos de cada paciente de manera individualizada.

Sección 5

5. CONCLUSIÓN

Los hallazgos obtenidos de este estudio respaldan la efectividad de la estrategia utilizada, evidenciada por la precisión lograda en la clasificación de las señales EEG. Se registraron 231, 285 y 242 predicciones correctas para las clases 3, 4 y 5, respectivamente, según la matriz de confusión. Sin embargo, se identificaron ciertos errores de clasificación que requieren atención especial, por ejemplo, para la Clase 1 se registraron 17 instancias mal clasificadas, de las cuales 2 se clasificaron como clase 3 y 15 como clase 4.

Las curvas ROC revelaron áreas de desempeño sólido con AUC prometedores, destacando especialmente la capacidad excepcional del modelo para distinguir la Clase 0, con valores de AUC de 0.94 en testeo y 0.92 en validación.

Asimismo, las métricas de evaluación, como exactitud, precisión, recuperación y F1-score, ofrecieron una comprensión detallada del desempeño del modelo por clase. Se observó una consistencia general en el rendimiento, aunque con variaciones entre clases específicas. Por ejemplo, para la Clase 1 se obtuvo una precisión de 0.73 en testeo y 0.70 en validación, y una recuperación de 0.93 en testeo y 0.95 en validación.

Al evaluar el modelo por paciente, se encontró una variabilidad considerable en la precisión entre individuos. Paciente 3 y Paciente 4 mostraron una alta precisión, con valores de 0.84 y 0.79 respectivamente, mientras que Paciente 7 y Paciente 8 obtuvieron una precisión notablemente menor, con valores de 0.22 y 0.26 respectivamente.

Estos resultados respaldan la efectividad de la estrategia implementada, pero también subrayan la necesidad de mejorar la capacidad del modelo para clasificar con mayor precisión los datos individuales de cada paciente y distinguir entre ciertas clases. Por tanto, se recomienda una validación más exhaustiva y una investigación adicional para optimizar este enfoque, utilizando estos resultados como base para futuros avances en el análisis de señales cerebrales mediante técnicas de aprendizaje automático.

Sección 6

6. TRABAJOS FUTUROS

Este algoritmo puede ser propicio para la continuación de proyectos futuros. En primer instancia, se prevé la implementación del modelo para el control del desplazamiento y dirección de una silla de ruedas, lo cual permite un avance hacia la automatización y la mejora de calidad de vida para personas que sufran enfermedades neurona-motoras o atrofias musculares. Sin embargo, el campo de acción del proyecto es bastante amplio por su gran relevancia y su potencial crecimiento en diversas aplicaciones donde se puede aplicar un algoritmo predictivo de señales electroencefalográficas en proyectos de investigación.

7. DISCUSIÓN

La precisión promedio de clasificación obtenida en este estudio se encuentra en un rango comparable con otros trabajos recientes que aplican SVM y otros clasificadores convencionales para decodificación de tareas motoras imaginadas a partir de señales EEG (Roy et al., 2018; Zhang et al., 2019; Yang et al., 2020). Si bien nuestros resultados muestran una dispersión mayor (desviación estándar de ± 0.224) en comparación con estudios como el de Yang et al. (2020) que obtuvo una desviación de ± 0.096, y Zhang et al. (2019) con ± 0.071, esto puede deberse al menor tamaño muestral y número de clases utilizado. Cabe resaltar que la variabilidad en la precisión entre sujetos también ha sido reportada en la literatura, y es esperable dadas las diferencias individuales en los patrones de actividad cerebral. Con un conjunto de datos más extenso y la aplicación de técnicas avanzadas como redes neuronales profundas, existe un amplio potencial para mejorar la precisión y alcanzar el estado del arte, como lo demuestran recientes investigaciones con LSTM y CNN que logran precisiones sobre 90 % en tareas similares de clasificación EEG.

8. OBSERVACIONES

8.1 Evaluación Sujeto no especifico

La realización de la evaluación del modelo utilizando el método sujeto no especifico se enfrentó a desafíos significativos relacionados con la capacidad computacional y el tiempo requerido. La complejidad y el tamaño del conjunto de datos junto con la complejidad algorítmica del método en cuestión resultaron en un tiempo de ejecución extremadamente largo, que estimamos en varias semanas. Esta prolongada duración no solo excedía los recursos disponibles en el equipo, sino que también planteaba limitaciones prácticas en términos de la viabilidad y la utilidad de los resultados obtenidos. Dada esta situación, se decidió interrumpir la ejecución de la evaluación, buscando alternativas más eficientes que pudieran proporcionar resultados significativos en un tiempo razonable y dentro de los límites de capacidad de hardware disponibles.

8.2 CRONOGRAMA

La planificación inicial del proyecto contemplaba un cronograma que abarcaba desde enero hasta junio, con la intención de llevar a cabo la recolección de señales mediante el uso del componente CYTON DONGLE, integrado en el casco Mark IV de la empresa openBCI. Lamentablemente, durante la ejecución del proyecto, se produjo un inconveniente significativo al dañarse dicho componente. Esta situación resultó en una interrupción prolongada en la recolección de señales cerebrales, ya que la adquisición de un nuevo CYTON DONGLE suponía un desafío financiero considerable para el proyecto.

La necesidad de reunir los fondos suficientes para la adquisición de este componente específico representó un obstáculo adicional, lo cual demandó un tiempo considerable hasta poder completar la compra. En consecuencia, se requirió una actualización del cronograma del proyecto, reajustando las fechas planificadas para llevar a cabo la recolección de datos desde agosto hasta diciembre. Este cambio fue necesario para acomodar la nueva planificación y retomar las actividades del proyecto una vez se solventó la situación relacionada con el componente dañado.

Referencias

Alazrai, R., Abuhijleh, M., Alwanni, H., & Daoud, M. I. (2019). A deep learning framework for decoding motor imagery tasks of the same hand using EEG signals. *IEEE Access*, *7*, 109612-109627.

Aldás, M. R. B., et al. (2018). *Estudio y análisis de métodos para la extracción de características y clasificación de emociones basados en eeg* [Tesis doctoral, Master's thesis].

Ali, H., Popescu, D., Hadar, A., Vasilateanu, A., Popa, R. C., Goga, N., & Hussam, H. A. D. Q. (2021). EEG-based Brain Computer Interface Prosthetic Hand using Raspberry Pi 4. *IJACSA) International Journal of Advanced Computer Science and Applications*, *12*(9).

Appriou, A., Cichocki, A., & Lotte, F. (2020). Modern machine-learning algorithms: for classifying cognitive and affective states from electroencephalography signals. *IEEE Systems, Man, and Cybernetics Magazine*, *6*(3), 29-38.

Arjunan, S. P., & Kumar, D. K. (2010). Decoding subtle forearm flexions using fractal features of surface electromyogram from single and multiple sensors. *Journal of neuroengineering and rehabilitation*, *7*, 1-10.

Artoni, F., Delorme, A., & Makeig, S. (2018). Applying dimension reduction to EEG data by Principal Component Analysis reduces the quality of its subsequent Independent Component decomposition. *NeuroImage*, *175*, 176-187.

Baranowski, J., Piątek, P., & Niemczyk, K. (2016). Design and implementation of non-integer band pass filters for EEG processing. *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 625-629.

Barrios Arce, J. I. (2019, julio). La matriz de confusión y sus métricas. https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Betancourt, G. A. (2005). Las máquinas de soporte vectorial (SVMs). *Scientia et technica*, *1*(27).

Cai, Q., & et al. (2021). A novel EEG seizure detection method based on convolutional neural network and support vector machine. *Biomedical Signal Processing and Control*, *61*, 102008.

Cantarelli, T. L., Júnior, J., & Júnior, S. (2016). Fundamentos da medição do eeg: Uma introdução. *Semin. ELETRONICA E AUTOMAÇÃO, Ponta Grossa*.

Cardona-Álvarez, Y. N., Álvarez-Meza, A. M., Cárdenas-Peña, D. A., Castaño-Duque, G. A., & Castellanos-Dominguez, G. (2023). A Novel OpenBCI Framework for EEG-Based Neurophysiological Experiments. *Sensors*, *23*(7), 3763.

Correlación y Covarianza [Accesed: 2023-10-23]. (s.f.).

Eng, M., John. (s.f.). ROC Analysis. Web-based Calculator for ROC Curves [Johns Hopkins University School of Medicine].

Espinosa, L. (2016). Electroencefalografía Inalámbrica: Una Mirada actual y propuesta de Sistema portátil. *Universidad Técnica Federico Santa María, Chile*.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861-874.

Granata, G., Di Iorio, R., Miraglia, F., Caulo, M., Iodice, F., Vecchio, F., Valle, G., Strauss, I., D'anna, E., Iberite, F., et al. (2020). Brain reactions to the use of sensorized hand prosthesis in amputees. *Brain and behavior*, *10*(11), e01734.

Liu, Y., Wang, Y., & Li, Y. (2021). EEG-based detection of Alzheimer's disease using wavelet transform and support vector machine.

OpenBCI/Tablero de ganglios [Accesed: 2023-03-15]. (s.f.).

OpenBCI/Ultracórtex Mark IV [Accesed: 2023-03-15]. (s.f.).

Phinyomark, A., Phukpattaranont, P., & Limsakul, C. (2012). Feature reduction and selection for EMG signal classification. *Expert systems with applications*, *39*(8), 7420-7431.

Roy, R., Sikdar, D., & Mahadevappa, M. (2020). Chaotic behaviour of EEG responses with an identical grasp posture. *Computers in Biology and Medicine*, *123*, 103822.

Roy, R., Sikdar, D., Mahadevappa, M., & Kumar, C. (2018). A fingertip force prediction model for grasp patterns characterised from the chaotic behaviour of EEG. *Medical & biological engineering & computing*, *56*, 2095-2107.

Sanabria Hernández, L. V., et al. (s.f.). Extracción de características y métodos de clasificación para reconocimiento de movimientos de mano a partir de señales de EMG y EEG: Revisión.

Soangra, R., Smith, J. A., Rajagopal, S., Yedavalli, S. V. R., & Anirudh, E. R. (2023). Classifying Unstable and Stable Walking Patterns Using Electroencephalography Signals and Machine Learning Algorithms. *Sensors*, *23*(13), 6005.

Tkach, D., Huang, H., & Kuiken, T. A. (2010). Study of stability of time-domain features for electromyographic pattern recognition. *Journal of neuroengineering and rehabilitation*, *7*(1), 1-13.

Too, J., Abdullah, A. R., & Saad, N. M. (2019). Classification of hand movements based on discrete wavelet transform and enhanced feature extraction. *International Journal of Advanced Computer Science and Applications*, *10*(6).

Wang, G., Zhang, Y., Wang, J., et al. (2014). The analysis of surface EMG signals with the wavelet-based correlation dimension method. *Computational and mathematical methods in medicine*, *2014*.

Wifi shield [Accesed: 2023-10-10]. (s.f.).

Yang, J., Ma, Z., Wang, J., & Fu, Y. (2020). A novel deep learning scheme for motor imagery EEG decoding based on spatial representation fusion. *IEEE Access*, *8*, 202100-202110.

Zhang, G., Davoodnia, V., Sepas-Moghaddam, A., Zhang, Y., & Etemad, A. (2019). Classification of hand movements from EEG using a deep attention-based LSTM network. *IEEE Sensors Journal*, *20*(6), 3113-3122.