**Autenticación de usuarios basada en reconocimiento de patrones EEG**

El proceso de experimentación implicó la recopilación de un cohorte de bases de datos denominado ‘DataSet IIb’, ofrecido públicamente por el Instituto Fraunhofer para los Sistemas de Comunicación Abiertos, el *Intelligent Data Analysis Group* (Klaus-Robert Müller, Benjamin Blankertz), el campus Benjamin Franklin de la Universidad de Medicina de Berlin y el *Neurophysics Group* Gabriel Curio. En este, se incluyen diversas colecciones de valores enteros de 16 bits que representan el voltaje encefálico generado por nueve sujetos durante 159 experimentos[[1]](#footnote-1), cada uno con duración fija de dos segundos.

En dichas pruebas, se solicitó a los participantes imaginar el movimiento, ya sea de la mano izquierda o derecha, siendo evidente la presencia de dos clases de tareas mentales a las que se hará referencia en el presente documento como y . Continuando con la descripción de los datos disponibles, es menester indicar que, la recopilación de los valores antedichos se realizó a una frecuencia de muestreo de 250 Hz, empleando los canales 2, 3 y 4 de la zona centro de la corteza cerebral, misma que se etiqueta bajo la leyenda “C” de acuerdo con el estándar internacional 10-20. Todo lo anterior, se organiza en matrices de la forma con los índices representando el canal, número de muestra en el tiempo y número de experimento respectivamente.

La prueba central de este documento, desarrollada en el lenguaje de programación Python 3.7.3, consiste en la elaboración de un clasificador bayesiano capaz de distinguir entre el movimiento del sujeto y , así como el movimiento del sujeto y . Expresando lo anterior en términos menos ortodoxos, sería adecuado afirmar que, el algoritmo fabricado debe ser capaz de identificar al responsable de cada movimiento. Por practicidad se optó por realizar únicamente comparaciones binarias entre sujetos, aunque se pretende que en productos posteriores el clasificador sea capaz de reconocer la autoría de un patrón neuronal entre los nueve posibles sujetos.

Pasando a detalles de la implementación, es importante además destacar que, dado el hecho de que los datos citados inicialmente se distribuyen exclusivamente en el formato de compatibilidad para el ambiente MATLAB “.mat” se requirió el uso del módulo *scipy.io*, que permitió su importación y subsecuente transformación en una lista de listas. Adicionalmente se empleó *numpy* para la operación de matrices y la extracción de características como varianza, media y desviación estándar; *scypi.signal* para el filtrado y remuestreo y *scipy.stats.norm* para el cálculo de la funcion de densidad de probabilidad gaussiana

De forma general, el programa sigue un modelo de inteligencia artificial supervisado, requiriendo inherentemente de dos fases: una de entrenamiento con datos previa y debidamente etiquetados, a los cuales se ha hecho alusión previamente, y una de clasificación, en la que se introducen valores desconocidos para probar la efectividad del adiestramiento. Así pues, el software elaborado recibe como parámetros los números de sujeto con los que se pretende realizar dicha etapa de aprendizaje, así como los detalles con que se llevará a cabo tal procedimiento, tales como: la cantidad de experimentos a considerar[[2]](#footnote-2), la constante denotando la frecuencia de muestreo, las bandas de frecuencia y que entre las cuales se filtrarán los datos, la clase por la que se pretende distinguir a ambos sujetos y los canales de recopilación que se deben considerar.

Subsecuentemente, el algoritmo se encarga de llevar a la memoria principal las bases de datos de y obteniendo, de acuerdo con lo estipulado inicialmente, dos matrices y , las cuales poseen un formato inteligible por Python. En ellas, se procede a aplicar un filtro Butterworth paso banda que emplea empíricamente el coeficiente 4 y las bandas de frecuencia y proporcionadas. Después de ello y, por cuestiones de seguridad, el software calcula el valor mínimo entre el número de experimentos de , y la variable . De lo contrario, es posible solicitar un entrenamiento con más datos de los disponibles en las colecciones. Dicho valor sobrescribe a , perdiendo su magnitud inicial.

Subsecuentemente, el algoritmo realiza la extracción de características, calculando la varianza de todos los elementos de y Los resultados se almacenan en dos nuevas matrices independientes y cuyas dimensiones son , donde representa el número de elementos en y el número de elementos en . Por cuestiones de practicidad, se indicarán las dimensiones antedichas, simplemente como puesto que el valor uno es irrelevante.

El clasificador recibe los siguientes parámetros: nombre del sujeto 1, nombre del sujeto 2, canales a considerar durante la tarea de clasificación, frecuencia de muestreo, cantidad de datos a usar en la fase de entrenamiento y duración del experimento

El algoritmo de entrenamiento, en primera instancia, calcula el mínimo entre los experimentos de entrenamiento que especificó el usuario, los experimentos del sujeto a y los experimentos del sujeto b para evitar un desequilibrio de los datos, garantizando así la equiprobabilidad de clases

Después se procede a filtrar los datos. Se aplica un filtro paso bandas

El experimento se repitió probando todas las posibles combinaciones

**Resultados**

1. Las bases de datos originales mostraban variabilidad en el número de experimentos. Para evitar que dicha inequidad afectase el proceso de clasificación, se redujeron todas las colecciones a 159 experimentos. [↑](#footnote-ref-1)
2. Es menester introducir esta variable para evitar el sobre entrenamiento del clasificador. Posteriormente en el documento se exhibirá con mayor detalle dicha problemática [↑](#footnote-ref-2)