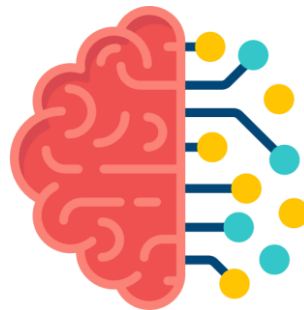


PROYECTO BCI

APRENDIZAJE DE MÁQUINA

NOVIEMBRE, 2023



HECHO POR:

IVÁN ALEJANDRO RAMOS HERRERA

INSTITUTO DE INVESTIGACIONES EN
MATEMÁTICAS APLICADAS Y SISTEMAS

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Proyecto BCI

Introducción

En la era actual de la ciencia y la tecnología, las interfaces cerebro-computadora (ICC) han emergido como un campo de investigación fascinante con el potencial de transformar la forma en que interactuamos con la tecnología. En este contexto se han desarrollado distintas áreas de investigación para abordar la creación y optimización de este tipo de herramientas; por ejemplo, con el uso de modelos de *Machine Learning* que sepan identificar patrones dentro de la actividad cerebral de una persona, a través de señales encefalográficas (*EEG*). Los enfoques líderes en la clasificación de datos *EEG* incluyen *Redes Neuronales Convolucionales* y *Recurrentes*, destacando por su capacidad para extraer características complejas y mejorar la precisión. La fusión de información multimodal, combinando datos *EEG* con otras modalidades biomédicas, es una de tantas estrategias eficaces para potenciar el rendimiento. Además, técnicas como la descomposición en valores singulares y la transformada wavelet siguen siendo relevantes para extraer características clave en los datos *EEG*. Por ejemplo, en el artículo *Federated Transfer Learning for EEG Signal Classification*, se propone la arquitectura *Federated Transfer Learning (FTL)* para abordar limitaciones de privacidad y escasez de datos. Utiliza la matriz de covarianza y técnicas de adaptación de dominio para lograr una precisión superior en comparación con otras arquitecturas en simulación mental de movimientos y ausencia de datos de múltiples sujetos.

En este contexto, la motivación detrás del presente trabajo surge de la necesidad de abordar las limitaciones identificadas en un artículo de 2023 encontrado en la literatura que nos proporcionó su conjunto de datos; realizado por *Joél Espinoza* en su investigación titulada "*ANÁLISIS COMPARATIVO DE ALGORITMOS EN LA CLASIFICACIÓN DE INTENCIÓN DE MOVIMIENTO MEDIANTE UNA INTERFAZ CEREBRO – COMPUTADORA*". En dicho trabajo se recopilaban señales cerebrales de varios individuos y se comparó el desempeño de modelos de clasificación utilizando dichos datos. Si bien se realizó un valioso análisis comparativo entre algoritmos, se identificaron deficiencias en el tratamiento de datos y en la evaluación de modelos.

La justificación de este reporte se centra en las deficiencias identificadas en la investigación previa de Espinoza, donde la falta de verificación de la integridad del conjunto de datos, el manejo inadecuado de clases desbalanceadas y la evaluación limitada de modelos constituyen obstáculos significativos para la validez y aplicabilidad de los resultados obtenidos; dejando espacio para construir una metodología de tratamiento para el conjunto de datos de señales cerebrales recolectado y trabajado por el trabajo antes descrito.

Desarrollamos un diseño experimental que aporta al método de Espinoza un re-muestreo al conjunto de datos, un filtrado de banda a las señales (para una extracción de características) y un ventaneo temporal de los datos. Dicho método de tratamiento aportaría a la claridad de las medidas de clasificación utilizadas por los modelos. Lo que nos permitió conocer métricas distintas en la evaluación de tres modelos implementados en un sistema con ***Python 3.11: XGBoost, Random***

Forest y una **Red Neuronal Recurrente LSTM** que con entrenamiento exhaustivo nos llevó encontrar exactitudes de predicción de hasta el **74%**, con recall de **76%** y F1-score de **73%**.

Punto de partida

La publicación *ANÁLISIS COMPARATIVO DE ALGORITMOS EN LA CLASIFICACIÓN DE INTENCIÓN DE MOVIMIENTO MEDIANTE UNA INTERFAZ CEREBRO – COMPUTADORA* de Joel Espinoza recopiló lecturas encefalográficas con una diadema de lectura cerebral de marca *Emotiv*, que registra catorce nodos cerebrales a una frecuencia máxima de 256 Hz por segundo. El conjunto de datos creado (y el cual es la base de este trabajo) consiste en una serie de cincuenta y cuatro lecturas encefalográficas de veinticinco segundos cada una, de veintisiete personas distintas (dos lecturas por persona). En donde cada lectura contiene los registros cerebrales de los individuos con los siguientes conceptos en su mente (cinco segundos por concepto, aproximadamente): 1. *Árbol, Cuaderno, Computadora, Perro, Nada* y 2. *Mover Mouse hacia; arriba, abajo, izquierda, derecha, Nada*.

Este conjunto se recopiló con la intención de servir como entrenamiento para Modelos de Machine Learning basados en un problema de clasificación multi-clase: *Support Vector Machines, Naive Bayes y Random Forest*.

Las deficiencias encontradas en el trabajo:

1. No se realizó ninguna verificación de la integridad del conjunto de lecturas, en pos de identificar errores existentes en:
 - a. Clases desbalanceadas.
 - b. Lecturas que contienen una cantidad ínfima de registros; lo que indicaría que no se realizó la captura correctamente.
2. No se realizó ninguna técnica de remuestreo para la corrección de irregularidades en la cantidad de lecturas para cada etiqueta (clase).
3. Se entrenaron los Modelos utilizando registros individuales (en donde cada segundo de lectura está formado por alrededor de 256 registros), es decir, cada muestra utilizada en el aprendizaje consiste en la fracción 256 de un segundo (1/256).
4. En la utilización de los datos no se consideró la temporalidad de los mismos; característica que nos indica que por naturaleza, un dato depende de los anteriores, y cada dato siguiente depende de lo que viene detrás en el tiempo.
5. En la evaluación de Modelos se utilizaron únicamente DOS medidas (*Accuracy y Desviación Estándar*); no son suficientes para medir el desempeño de los mismos.

Metodología y Diseño Experimental

Preprocesamiento: Se tomó el conjunto original del trabajo de *Espinoza*, que consiste en dos *datasets* (para “*evocación de concepto*” y para “*intención de movimiento*”) para cada individuo (de entre 27 individuos).

1. Exploración. Objetivo: Identificar irregularidades en los datos:

- a. Recopilar los cincuenta y cuatro conjuntos de lecturas y realizar un conteo de clases para los mismos.
- b. Calcular un índice de balance en la cantidad de clases de cada conjunto. Índice de desbalance = ($\#$ clase mayoritaria / $\#$ clase minoritaria).

2. Remuestreo. Objetivo: Eliminar irregularidades en los datos:

- a. A aquellos conjuntos con un índice de desbalance mayor a 2:
 - i. Se toma la clase mayoritaria (clase con más registros) y se hace un recorte eliminando los registros (desde arriba hacia abajo para tratar de minimizar el ruido en los inicios de lectura para cada etiqueta), de manera que se eliminen los registros hasta tener una cantidad correspondiente a cinco segundos aproximadamente (con 256 registros por segundo).
 - ii. Se mide nuevamente el desbalance comprobando si el promedio de lecturas para cada etiqueta corresponde a un aproximado de 1024 lecturas (cinco segundos a 256Hz por segundo).
 - iii. Para aquellos conjuntos de datos cuyas cantidades de muestra están alejadas del promedio de registros normal se reevalúa bajo dos estrategias: 1. Si las lecturas promedio están muy por debajo de lo requerido, se elimina el dataset; 2. Si la clase mayoritaria sigue estando por encima del promedio, se remuestrea nuevamente.
- b. Se mide nuevamente que los conjuntos contienen aproximadamente 1024 registros por cada etiqueta, para cada una de las etiquetas; es decir, que se tienen cinco segundos de lectura para cada individuo en cada clase.

3. Transformación. Objetivo: Transformar los datos según su naturaleza.

- a. Aplicar remuestreo de los datos con base en filtrarlos mediante un *pasa bandas*:
 - i. Se aplicó un filtro de pasa bandas de tipo *Butterworth* con cortes entre 12Hz y 50Hz (rango de frecuencias a señales cerebrales encefalográficas EEG *Beta* y *Gamma*; asociadas a actividades cerebrales de concentración y evocación de pensamientos e intenciones) con base al principio de *Nyquist-Shannon*, que establece que la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima de interés, en una señal que se ha digitalizado ($nyquist = 256 / 2$). Este filtro permite: Enfocar el problema en el rango de frecuencias de interés (para el problema de evocación de conceptos), reducir el ruido al digitalizar las señales y a modo de **extracción de características**, en donde los rangos de interés están en las ondas *Beta* y *Gamma*.
- b. Aplicar una Transformada Rápida de Fourier a los datos:
 - i. La FFT convierte los datos de EEG del dominio del tiempo al dominio de la frecuencia, lo que le permite visualizar el contenido de frecuencia de la señal en cada momento del tiempo.
- c. Aplicar normalización a los datos:

- i. Permite escalar los datos para que sus valores queden en rangos definidos, facilitando el procesamiento en los modelos.

4. Separación. Objetivo: Crear datasets con sólo los datos necesarios:

- a. Se separan en dos carpetas distintas los datasets para cada categoría:
 - i. Evocación de concepto (con las etiquetas: Árbol, Cuaderno, Computadora, Perro, Nada).
 - ii. Intención de movimiento (con las etiquetas: Mouse; arriba, abajo, izquierda, derecha, Nada).
- b. Se eliminarán las clases de la etiqueta (“Nada”) para ambas categorías. Este se consideró como una clase de control, y no como algo para ser realmente recopilado o analizado.
- c. Se reemplazarán los nombres de etiquetas por números:
 - i. Evocación Concepto: Árbol [0], Cuaderno [1], Computadora [2]. Perro [3].
 - ii. Intención de Movimiento: MouseUp [0], MouseDown [1], MouseLeft [2], MouseRight [3].

5. Ventaneo. Objetivo: Tratar los datos como ventanas de tiempo:

- a. Separación en ventanas de 0.12 segundos: De cada dataset y para cada conjunto de etiquetas, se tomarán 30 registros (equivalente aproximado a 0.12 segundos) y se tomará este conjunto de datos de 30 registros como una muestra de dato único. Es decir, cada muestra es equivalente a una octava parte de un segundo.

6. Modelos. Objetivo: Ya con los datos procesados, evaluar modelos de clasificación:

- a. Métricas para todos los Modelos: Accuracy, Recall, F1-Score Macro, y Precisión de predicción para cada clase, todas con K-Folds Cross Validation con k=5 y aplicadas a todos los sujetos de muestra.
- b. Modelos a probar:
 - i. **XGBoost Classifier:** Utilizando el clasificador desde la paquetería de Python de *Sklearn* y con los parámetros: **loss='log_loss', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse'**. Es un algoritmo de aprendizaje supervisado efectivo para clasificación y regresión. Utiliza de entrada un conjunto de datos de entrenamiento con características y etiquetas de clase o valores objetivo (en este caso matrices de 14x30 [cantidad de canales x ventana de tiempo]) con un vector y que tiene asociada una etiqueta [0, 1, 2, 3]. El algoritmo construye múltiples árboles de decisión débiles secuencialmente, corrigiendo los errores del modelo anterior. La salida es un modelo predictivo que puede hacer predicciones precisas en nuevos datos, ya sea para clasificación o regresión. Este algoritmo de salida recibe la entrada ya descrita y calcula como salida una de las etiquetas [0, 1, 2, 3].
 - ii. **Random Forest Classifier:** Utilizando el clasificador desde la paquetería de Python de *Sklearn* y con los parámetros: **n_estimators=100, criterion='gini', bootstrap=True**. Algoritmo de aprendizaje supervisado versátil y robusto, especialmente eficaz para problemas de clasificación y regresión. Similar a XGBoost, requiere un conjunto de datos de entrenamiento con características y

etiquetas de clase o valores objetivo. En el contexto descrito, el conjunto de datos de entrada sería una matriz de dimensiones 14x30, representando la cantidad de canales y la ventana de tiempo, respectivamente. Random Forest construye múltiples árboles de decisión de forma independiente y combina sus predicciones para mejorar la precisión general del modelo. La salida es una predicción que puede ser una etiqueta de clase específica (entre [0, 1, 2, 3]).

iii. **Red Neuronal Recurrente LSTM:** Con las siguientes características:

1. **Arquitectura:**

a. **Capa LSTM:**

- i. **Tamaño de entrada:** 14 (n secuencias de tamaño 60 extraídas del ventaneo de datos con 14 características; matrices de 60x14).
- ii. **Tamaño del estado oculto:** 64 neuronas.
- iii. **Número de capas apiladas:** 2
- iv. **Batch_first:** True.
- v. **Salida:** out[:, -1, :] (último paso de tiempo).

b. **Capa FeedForward:**

- i. **Tamaño de entrada:** 64 (salida de LSTM).
- ii. **Tamaño de salida:** 4 (para cada clase).
- iii. **Función de salida:** Arg Max: Softmax Argumento del valor máximo de un vector de 4 clases.

2. **Criterio de costo:** Función de Entropía Cruzada.

3. **Optimizador:** Adaptive Momentum (ADAM).

4. **Tasa de aprendizaje:** 0.001.

5. **Épocas de entrenamiento:** 300.

La red neuronal recurrente LSTM se ajusta a los datos mediante el procesamiento de secuencias temporales de tamaño 60, representando 14 características extraídas de las ventanas de datos. Con dos capas LSTM apiladas, la red aprende patrones temporales, y su salida en el último paso de tiempo alimenta una capa FeedForward. Esta capa utiliza la función Arg Max con Softmax para asignar probabilidades a las clases. Este modelo busca capturar y generalizar patrones complejos en los datos EEG, mejorando la capacidad de clasificación de los datos de señales.

Experimentación previa a los resultados

Para la identificación del impacto de los distintos pre-procesamientos aplicados sobre el conjunto de datos se tomaron los **dos conjuntos (evocación de concepto e intención de movimiento)** de **UN solo sujeto**, para probar la capacidad de predicción de los modelos **utilizando en esta primera instancia sólo la medida de exactitud (accuracy)** promediado para cada uno de los **5 folds** empleados con **KFolds-CrossValidation**. **EN LA RED NEURONAL (RNN) SE UTILIZARON SÓLO 100 ÉPOCAS DE ENTRENAMIENTO EN ESTE PASO.**

Para el Conjunto 1 [Evocación de Concepto]:

Pre-Procesamiento de Dataset	Accuracy		
	KGBoost	RandForest	RNNLSTM*
Original	59.45%	59.85%	61.75%
Original – Balanceado – Filtrado	17.04%	11.11%	31.85%
Original – Balanceado – Filtrado – Fourier	67.47%	62.95%	71.85%
Original – Balanceado – Filtrado – Fourier - Normalizado	67.47%	64.44%	21.48%

Para el Conjunto 2 [Intención de Movimiento]:

Pre-Procesamiento de Dataset	Accuracy		
	KGBoost	RandForest	RNN LSTM
Original	38.12%	36.00%	37.15%
Original – Balanceado – Filtrado	36.52%	30.77%	41.05%
Original – Balanceado – Filtrado – Fourier	75.36%	77.80%	67.95%
Original – Balanceado – Filtrado – Fourier - Normalizado	75.36%	76.15%	32.82%

Es entonces que como propuesta de tratamiento de los datos se propone: **Original – Balanceado – Filtrado – Fourier**. Este método se utilizó para medir resultados de un paso final de evaluación.

Resultados

En este paso se utilizó la configuración de procesamiento **Original – Balanceado – Filtrado – Fourier**, y **SE TOMARON TODOS LOS DATASETS (27 sujetos)** y se entrenó cada modelo con **KFolds Cross Validation** con el valor de **5 folds**, y para cada dataset se aplicaron las métricas **Accuracy, Recall, F1-Score-Macro y Precisión (para cada clase)**, de manera que se **promedió cada métrica para los 5 folds** y estos valores se **promediaron entre todos los datasets**. Obteniendo los siguientes resultados.

Para el Conjunto 1 [Evocación de Concepto]:

MODELO	Accuracy	Recall Macro	F1-Macro	Precisión	
XGBoost	72.26%	72.47%	70.25%	Árbol	74.45%
				Cuaderno	67.70%
				Computadora	75.24%
				Perro	74.14%
Random Forest	68.76%	69.00%	66.15%	Árbol	69.37%
				Cuaderno	62.94%
				Computadora	71.82%
				Perro	70.86%
RRN LSTM	74.77%	76.29%	73.89%	Árbol	95.26%
				Cuaderno	58.09%
				Computadora	94.99%
				Perro	56.28%

Para el Conjunto 2 [Intención de Movimiento]:

MODELO	Accuracy	Recall Macro	F1-Macro	Precisión	
XGBoost	72.60%	73.52%	71.45%	MouseUp	76.40%
				MouseDown	72.61%
				MouseLeft	74.53%
				MouseRight	73.35%
Random Forest	70.43%	71.83%	68.83%	MouseUp	75.11%
				MouseDown	66.98%
				MouseLeft	71.16%
				MouseRight	72.55%
RRN LSTM	74.19%	75.48%	73.20%	MouseUp	92.05%
				MouseDown	56.48%
				MouseLeft	91.59%
				MouseRight	58.32%

Discusión de resultados

Los resultados obtenidos muestran que la configuración de procesamiento "**Original – Balanceado – Filtrado – Fourier**" da la mejor capacidad de clasificación con el conjunto de datos, y al mismo tiempo solucionando las deficiencias observadas en el trabajo de *Joél Espinoza*. El estudio de ablación, al analizar diferentes configuraciones de pre-procesamiento, destaca la importancia de equilibrar las clases, aplicar filtrado en el dominio de la frecuencia y normalizar los datos para obtener un rendimiento óptimo, además de que demostró que nuestra mejor propuesta de modelo radica en utilizar una **Red Neuronal Artificial Recurrente LSTM** con la arquitectura antes descrita, pero cambiando el parámetro de **Épocas de entrenamiento** a **300**.

Se descubrió que la **RNN LSTM** tuvo un rendimiento excepcional en la clasificación, especialmente en la captura de patrones temporales en las señales EEG. La combinación de balanceo de clases, filtrado en el dominio de la frecuencia y normalización contribuyó significativamente a la mejora de la precisión de los modelos. En comparación con la investigación previa de *Joél Espinoza*, nuestra propuesta aborda las deficiencias identificadas, como el desbalance de clases y la falta de consideración de la temporalidad de los datos, quien con su propia configuración de **Datos Originales – Fourier**, propone evaluaciones de modelos poco útiles.

Desventajas y Limitaciones

A pesar de los resultados positivos, la propuesta tiene algunas limitaciones. La capacidad de generalización puede verse afectada por la variabilidad interindividual en las señales cerebrales y por la diferencia entre la actividad cerebral de los sujetos; de modo que el entrenamiento de un modelo destinado a predecir sobre la actividad real de un individuo tiene que ser (en este contexto y sin herramientas externas de adaptación de dominio), tiene que ser entrenado exclusivamente con lecturas del mismo individuo.

Además, la complejidad computacional de la **RNN LSTM** puede limitar su aplicabilidad en entornos con recursos limitados, al menos para el entrenamiento, y si se desea construir un sistema que en tiempo real haga predicciones; el trabajo necesario para la adaptación (y sí, los recursos del sistema que los empleará) quedan fuera de este trabajo, pero se concluye como importante.

Conclusiones

La configuración de procesamiento propuesta ofreció una más clara metodología evaluada que demuestra brindar buenas capacidades en la implementación de modelos en herramientas de *BCI*. El tratamiento de las señales en datos capturados en la investigación de la que parte este trabajo, configurada como **Original – Balanceo – Filtrado – Fourier**, con el modelo **RNN LSTM** demostró ser una herramienta poderosa para la clasificación de señales *EEG*, capturando patrones temporales con alta precisión.

Este trabajo sienta las bases para futuras investigaciones y desarrollos en el campo de las *BCI*. Se reconoce la necesidad de abordar las limitaciones identificadas, como la adaptación interindividual y la complejidad computacional, para avanzar hacia aplicaciones prácticas en entornos del mundo real. El camino hacia sistemas más adaptables y eficientes en recursos se presenta como un desafío fascinante para el futuro de las interfaces cerebro-computadora.

Referencias

- Espinoza Sánchez, J. A. 2023. ANÁLISIS COMPARATIVO DE ALGORITMOS EN LA CLASIFICACIÓN DE INTENCIÓN DE MOVIMIENTO MEDIANTE UNA INTERFAZ CEREBRO – COMPUTADORA.
Recuperado de <https://github.com/Joul24py/UAA-ICI-Thesis>.
- Ju, C., Gao, D., Mane, R., Tan, B., Liu, Y., Guan, C. (2021). Federated Transfer Learning for EEG Signal Classification. arXiv. Recuperado el 20 de noviembre del 2023 de: <https://arxiv.org/abs/2004.12321>.
- Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. University of Washington. Recuperado el 10 de Octubre del 2023 de: <https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf>.
- Breiman, L. (2001). RANDOM FORESTS. University of California. Recuperado el 10 de Octubre del 2023 de: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>.
- Guetschel, P., Tangermann, M. (2023). Transfer Learning between Motor Imagery Datasets using Deep Learning - Validation of Framework and Comparison of Datasets. arXiv. Recuperado el 22 de noviembre de: <https://arxiv.org/pdf/2311.16109.pdf>.
- Staudemeyer, R., Rothstein, E. (2019). – Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks Faculty of Computer Science Schmalkalden University of Applied Sciences. Recuperado el 10 de Octubre del 2023 de: <https://arxiv.org/pdf/1909.09586.pdf>.