

DESEMPEÑO COMPARATIVO DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO Y PROFUNDO EN LA CLASIFICACIÓN DE ACTIVIDAD MUSCULAR USANDO EMG DEL BÍCEPS

Luis Alberto Retamozo Rodriguez - luisretamozoar@gmail.com

Resumen – Se analizaron señales electromiográficas (EMG) del bíceps braquial, obtenidas de archivos .wav (2000 Hz, 1 canal, 96 sujetos), durante ejercicios de levantamiento de mancuernas con series de 10 repeticiones. El objetivo fue discriminar patrones electrofisiológicos característicos de los estados de reposo y actividad muscular. La compatibilidad de los registros se verificó, confirmando una frecuencia de muestreo uniforme de 2000 Hz, acorde con el teorema de Nyquist (mínimo 1000 Hz para señales EMG) y protocolos estándar de adquisición, asegurando su idoneidad para análisis en el dominio temporal.

Por su capacidad para capturar dinámicas temporales y de amplitud de la actividad muscular el procesamiento se centró en la extracción de características de Hjort (actividad, movilidad y complejidad) calculadas a partir de la envolvente de Hilbert suavizada de la señal EMG, mientras que el preprocesado combinó técnicas que incluye un filtrado de banda (20-150 Hz), un filtrado rechaza banda (60Hz, y su primer armónico 120Hz), cálculo de la envolvente de Hilbert y un filtro de media móvil, conservando una representación eficiente de los patrones electrofisiológicos lo cual a posteriori facilitó el etiquetado de estados musculares mediante el uso de K-means con un coeficiente de silueta de 0.6648 , indicando una separación consistente entre estados.

Para la evaluación del desempeño, se implementaron diversos modelos de clasificación: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) y una arquitectura híbrida CNN-LSTM. Estos modelos alcanzaron precisiones en el rango de 0.96 a 0.99, destacando el algoritmo Random Forest, que obtuvo una exactitud del 99 % para la clase de

actividad y del 98 % para la clase de reposo, con un tiempo de inferencia promedio de 0.0411 segundos. La exclusión de registros con artefactos y el uso de características de Hjorth demostraron ser estrategias efectivas para una clasificación robusta. Estos resultados evidencian el potencial del enfoque propuesto para aplicaciones clínicas y de rehabilitación, superando métodos tradicionales basados en umbrales y ofreciendo alta precisión y escalabilidad para el monitoreo en tiempo real de la actividad muscular.

Objetivos

Objetivo General: Evaluar el rendimiento de modelos Machine learnign frente a modelos Deep learning en la identificación de patrones en series temporales de señales electrofisiológicas teniendo en cuenta métricas de rendimiento como la exactitud (accuracy), la precisión (precision), el recall (tasa de verdaderos positivos), el F1 score, la tasa de falsos positivos, la curva ROC (Receiver Operating Characteristic), tiempo de procesamiento y costo computacional.

Objetivos Específicos

1. Seleccionar y preparar los conjuntos de datos de señales electrofisiológicas.
2. Desarrollar modelos de Machine Learning para la identificación de patrones electrofisiológicos.
3. Desarrollar modelos de Deep Learning para la identificación de patrones electrofisiológicos.
4. Comparar el rendimiento de los métodos y generar conclusiones sobre la efectividad de los modelos.

Metodología

La metodología implementada en este estudio se centró en el procesamiento, modelado

y evaluación de señales electromiográficas (EMG) del bíceps braquial, con el objetivo de comparar el desempeño de diversos modelos de aprendizaje automático (ML) y aprendizaje profundo (DL) en tareas de clasificación de actividad muscular, tal como se presenta en la figura 0.

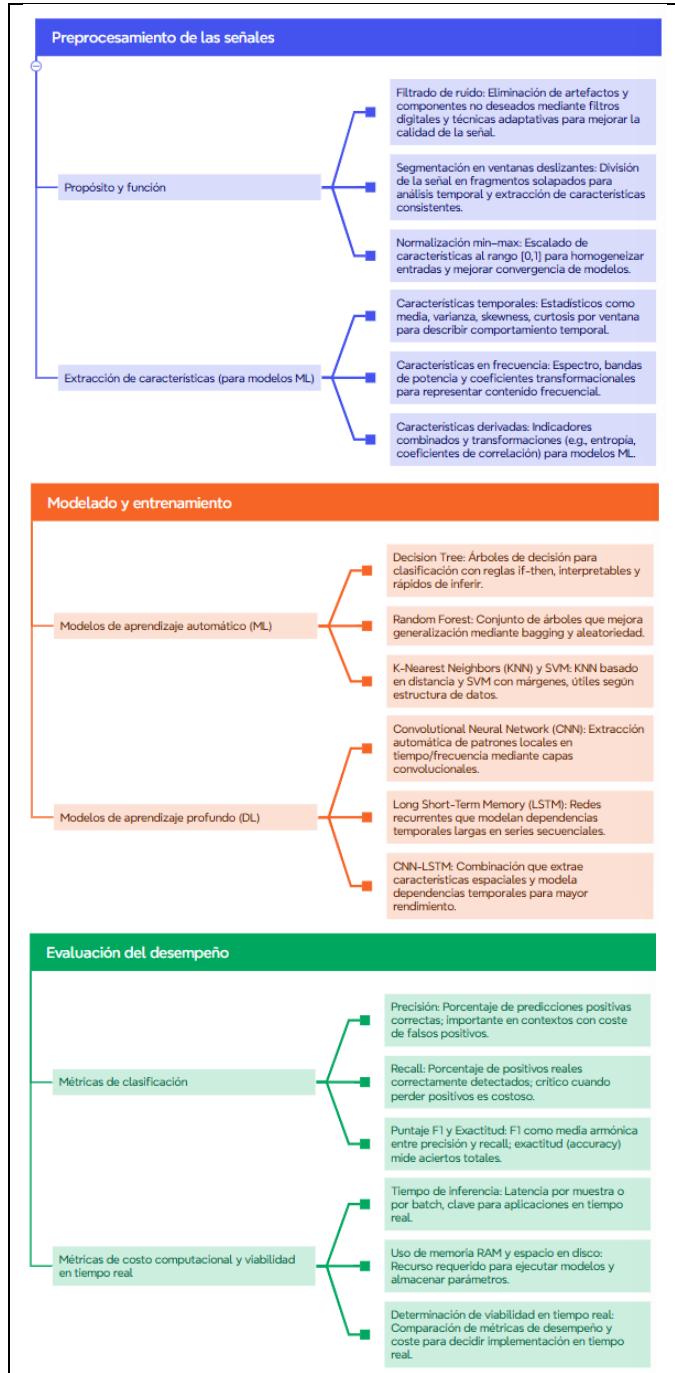


Fig 0: Metodología implementada.

Descripción de los datos

Los datos de forma de onda del conjunto de datos EMG-DATASET-CIARDEB-MAGMA se

recopilaron con electrodos superficiales mediante el uso del software Biometrics Datalite, los registros se seleccionaron y convirtieron en una base de datos estructurada orientada a su implementación en el entrenamiento de modelos aprendizaje automático y profundo. Las etiquetas correspondientes se almacenaron en un formato .csv que pueden procesarse fácilmente con software estándar.

Métodos de adquisición

1. Los datos de señal sin procesar se registraron y almacenaron en formato .wav, proporcionando información de registros electromiográficas (EMG) de bíceps y tríceps durante ejercicios de levantamiento de mancuernas que involucran el centro de gravedad, el oblicuo externo e interno, el recto anterior y la cresta ilíaca.
2. Para la generación de los metadatos se seleccionó una derivación única correspondiente al bíceps braquial con electrodos de referencia en el epicóndilo lateral del húmero.
3. Los metadatos generales correspondientes: actividad, movilidad, complejidad y etiqueta de estado se recopilaron en el archivo emg_features_optimized.csv.
4. Cada registro fue analizado y categorizado utilizando el algoritmo K-Means, obteniéndose un coeficiente de silueta de 0.6648, lo que indica una estructura de agrupamiento moderadamente consistente.
5. Los datos crudos se organizaron de manera jerárquica para facilitar el acceso y el análisis de señales electromiográficas (EMG) recolectadas de músculos como el bíceps y tríceps braquial durante ejercicios funcionales,

en general, el conjunto de datos se distribuye de la siguiente manera:

EMG-DATASET-CIARDEB-MAGMA-MAIN-BICEP Y TRICEP	
Bicep y tricep	
archivos de audio	
BICEP	
Atletismo	
Baloncesto	
Futbol	
... Otros deportes	
TRICEP	
...	
archivos de texto	
BICEP	
...	
TRICEP	
...	

Se registraron señales EMG superficiales de 104 sujetos voluntarios, sin embargo, tras la inspección y eliminación de registros con artefactos o calidad insuficiente el conjunto de datos seleccionados solo comprende 96 registros de EMG con una única derivación en formato .wav correspondiente al músculo bíceps braquial, el uso del .wav para almacenar señales EMG se justifica por su capacidad para representar datos de onda de forma cruda y de alta resolución. Los criterios de selección se detallan en la Tabla 1, que resalta características demográficas de la población estudiada.

Característica	Descripción
Tipo de población	Atletas de alto rendimiento
Disciplinas	Fútbol, fútbol sala, baloncesto, atletismo, voleibol, porismo, taekwondo, softbol, patinaje, tenis de mesa, tenis de campo, fitness, rugby, karate do
Rango de edad	17-35 años
Género	Masculino y femenino

Tabla 1: Características de la Población Estudiada: Atletas de Alto Rendimiento.

Este grupo de individuos exhibe picos de rendimiento neuromuscular y es ideal para estudiar adaptaciones musculares sin las

variables confusoras del envejecimiento o inmadurez fisiológica, además la diversidad de disciplinas deportivas asegura una generalización de los hallazgos a múltiples contextos [1]. La distribución los voluntarios se detalla en a continuación en la tabla 2:

Disciplina	Número de Voluntarios
Atletismo	18
Baloncesto	8
Fútbol	7
Fútbol Sala	2
Karate	6
Patinaje	2
Porismo	10
Rugby	18
Softball	6
Tenis	1
Tenis de Mesa	6
Ultimate	9
Voleibol	12
Total	104

Tabla 2: Distribución de las Disciplinas Deportivas Practicadas por los Voluntarios.

Reorganización de los datos

Antes de iniciar con el preprocesamiento secuencial de los datos, se separaron los registros de interés del resto, y se dispusieron en un único directorio nombrado "000", para ello se codificaron los algoritmos de las funciones scanfolder y deepscan en el lenguaje de programación MATLAB utilizado durante esta etapa, ambas funciones (figura 1) se describen a continuación:

```

function [folderlist] = scanfolder(folder)
    subdirtemp = dir(folder);
    folderlist = [" "];
    for i = 3:length(subdirtemp)
        folderlist(i-2, 1) = fullfile(folder, subdirtemp(i).name);
    end
end

function [sublist] = deepscaan(folderlist)
    sublist = {};
    for i = 1:length(folderlist)
        temp_sublist = scanfolder(folderlist(i));
        for j = 1:length(temp_sublist)
            sublist{end + 1, 1} = temp_sublist(j);
        end
    end
    sublist = string(sublist);
end

```

Fig 1: Código de la función scanfolder y deepscan.

Scanfolder: recibe una cadena de caracteres con la ruta al dataset, utiliza dir y fullfile de MATLAB para listar y retornar un arreglo de rutas con todas las subcarpetas del directorio padre "emg-dataset-ciardel-magma-main Bicep y tricep/Bicep y tricep/archivos de audio/".

Deepscaan: escanea recursivamente una lista de carpetas, utilizando scanfolder para obtener las subcarpetas de cada carpeta en la lista y concatenarlas en un arreglo de cadenas, lo que permite rastrear el origen de los registros EMG asignados tal como se muestra en la figura 2.

REG1	emg-dataset-ciardel-magma-main-Bicep y tricep/Bicep y tricep/archivos de audio/BICEP/Atletismo/S10_ATL_BIC.wav
REG2	emg-dataset-ciardel-magma-main-Bicep y tricep/Bicep y tricep/archivos de audio/BICEP/Atletismo/S18_ATL_BIC.wav
REG3	emg-dataset-ciardel-magma-main-Bicep y tricep/Bicep y tricep/archivos de audio/BICEP/Atletismo/S1_ATL_BIC.wav
REG4	emg-dataset-ciardel-magma-main-Bicep y tricep/Bicep y tricep/archivos de audio/BICEP/Atletismo/S2_ATL_BIC.wav
REG5	emg-dataset-ciardel-magma-main-Bicep y tricep/Bicep y tricep/archivos de audio/BICEP/Atletismo/S3_ATL_BIC.wav
...	...

Fig 2: primeros cinco registros de la tabla de rutas.

Después de inspeccionar los datos, se excluyeron del análisis aquellos que presentaban artefactos o una cantidad excesiva de ruido, ya que estos contaminantes pueden distorsionar la extracción de características y reducir la precisión de los modelos de clasificación, los registros excluidos se detallan en la Tabla 3.

REG11	REG30	REG60
REG17	REG44	REG82
REG23	REG47	REG87

Tabla 3: Registros Electromiográficas excluidos del análisis debido a artefactos y ruido Excesivo.

Debido a que se realizó uso de registros en formato .wav, se asume que los valores de amplitud han sido cuantificados como enteros de 16 bits con signo, con un rango dinámico de -32,768 a +32,767. Por lo tanto, al utilizar la función audioread en MATLAB, la señal es automáticamente convertida al tipo double y normalizada al rango de -1 a 1 [2].

Preprocesado

De acuerdo con De Luca [1], el procesamiento de la señal electromiográfica (EMG) requiere considerar sus características intrínsecas para garantizar la calidad y fiabilidad de los datos a obtener. Siguiendo las recomendaciones establecidas en el artículo *Surface Electromyography: Detection and Recording*, se prestó especial atención a las características relevantes de la señal, detalladas en la Tabla 4. Asimismo, se aplicaron los pasos de preprocesamiento descritos en la Tabla 5, asegurando un tratamiento sistemático y riguroso de los datos para optimizar su análisis posterior.

No.	Característica
1	La amplitud de la señal EMG es de naturaleza estocástica.
2	Puede representarse razonablemente mediante una función de distribución gaussiana.
3	La energía utilizable de la señal se limita al rango de frecuencia de 0 a 500 Hz.
4	La energía dominante ocupa el rango de 50 a 150 Hz.

Tabla 4: Características Relevantes para este estudio de la Señal Electromiográfica (EMG).

Antes de detallar los pasos de preprocesamiento aplicados a la señal electromiográfica (EMG), es fundamental analizar su comportamiento en el dominio de la frecuencia. En la Figura 3 se presenta un ejemplo característico del espectro de frecuencia de la

señal EMG, que ilustra la distribución de su energía en el rango de 0 a 500 Hz tal como describe De Luca.

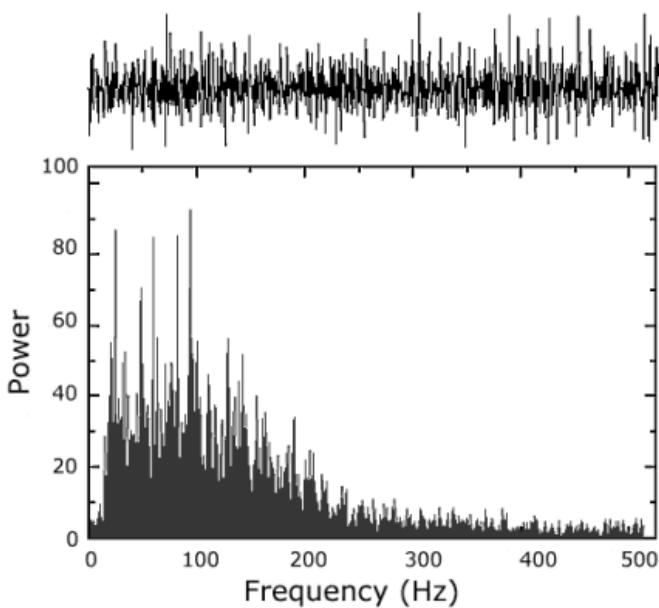


Fig 3: Espectro de frecuencias de la señal EMG detectada en el músculo tibial anterior durante una contracción isométrica de fuerza constante al 50 % del máximo voluntario, recuperada de [3].

Este espectrograma proporciona una representación visual útil para comprender las propiedades de la señal y fundamentar las etapas de preprocesamiento.

No.	Paso de Preprocesamiento
1	Filtro Notch a 60 Hz y su primer armónico (120 Hz) para eliminar interferencias eléctricas de la red.
2	Filtro pasa-banda Butterworth de sexto orden, con frecuencias de corte de 20 Hz y 150 Hz, para conservar las componentes relevantes de la señal EMG y eliminar ruido de baja y alta frecuencia.
3	Extracción de la envolvente mediante la transformada de Hilbert, lo que permite resaltar la amplitud instantánea de la señal rectificada.
4	Media móvil utilizando una ventana rectangular de 50 muestras (equivalente a 25 ms a una frecuencia de muestreo de 2000 Hz), para suavizar la envolvente y reducir fluctuaciones rápidas no deseadas.

Tabla 5: Pasos de Preprocesamiento de la Señal Electromiográfica (EMG).

A continuación, en la figura 4 se presentan imágenes representativas de la señal EMG en cada etapa de la cadena de preprocesamiento, destacando los efectos de la extracción de la envolvente Hilbertiana y el suavizado.

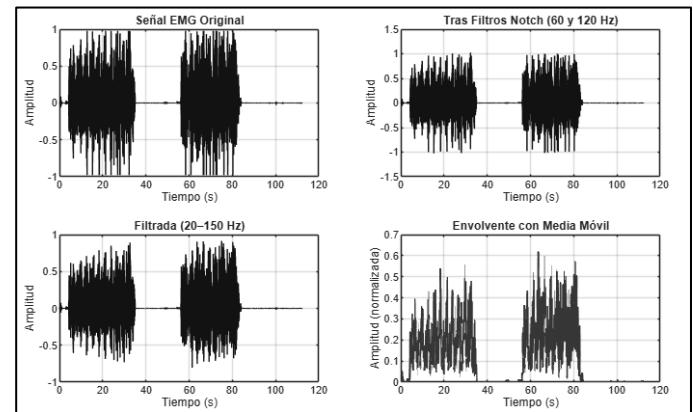


Fig 4: Cambio en la morfología de la señal EMG a través de las etapas de preprocesamiento.

Estas visualizaciones permiten comprender la evolución de la señal electromiografía a lo largo del preprocesamiento de la señal cruda resaltando componentes fisiológicamente relevantes.

Justificación de las etapas de preprocesado

Las señales EMG se procesaron con un filtro pasa-banda Butterworth de 6º orden (20–150 Hz), aplicado de manera bidireccional (filtfilt) para evitar desfase. Además, se emplearon filtros notch a 60 y 120 Hz para atenuar interferencia de red eléctrica y armónicos.

Filtro Notch a 60 Hz y 120 Hz: Las interferencias de red eléctrica constituyen la fuente principal de ruido en el EMG [4][5], por ello la implementación del filtro tipo Notch se utiliza tradicionalmente para reducir el ruido de la línea eléctrica [6], aunque al atenuar la componente de frecuencia correspondiente a estas frecuencias también se puede eliminar parte útil de la señal, la supresión de la interferencia de red puede justificar su uso, especialmente cuando se dispone de técnicas de filtrado bidireccional comofiltfilt de MATLAB que

minimizan la distorsión de fase, como se observa en las respuestas de fase en las Figuras 5.

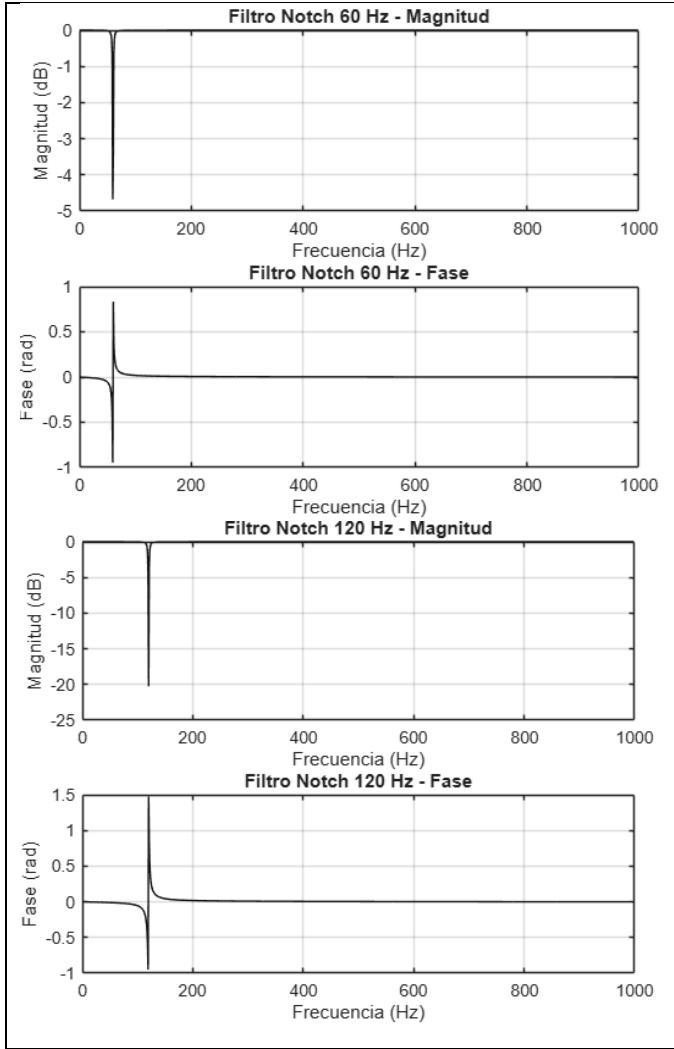


Fig 5: Respuestas en fase y frecuencias de los filtros Notch a 60 y 150 Hz.

Filtro Butterworth de 20–150 Hz: El filtro Butterworth tiene una respuesta en frecuencia monótona en la banda pasante, es decir, que no presenta ondulaciones, lo que preserva la linealmente la amplitud de la señal (como se ilustra en la respuesta de magnitud y fase en la Figura 6).

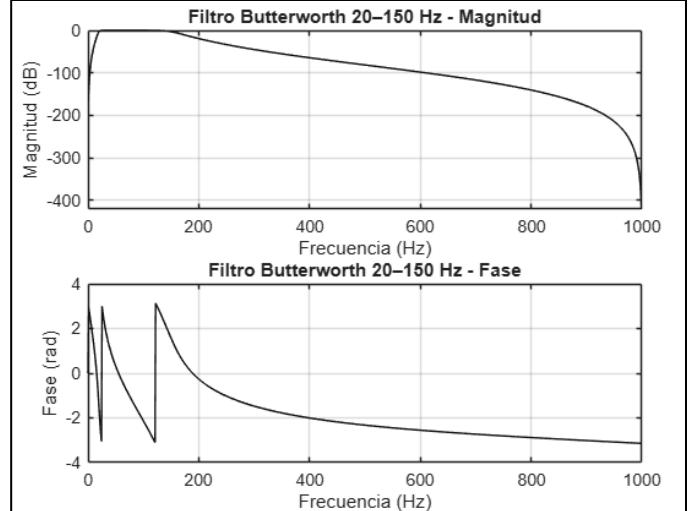


Fig 5: Respuestas en fase y frecuencias del filtro pasa banda de 20 a 150 Hz (sexto orden).

Esta característica lo hace ideal para el preprocesado de señales y su posterior análisis temporal, en esta caso, aunque la energía útil del EMG superficial se concentra en el rango 20–500 Hz y principalmente entre 50–150 Hz [7], se ha seleccionado el rango de 20–150 Hz teniendo en cuenta las consideraciones de De Luca en *Surface Electromyography: Detection and Recording*, de este modo se enfatiza la porción principal del espectro muscular, el límite inferior en 20 Hz es típicamente utilizado para eliminar artefactos de movimiento y baja frecuencia [5], mientras que el corte superior en 150 Hz descarta ruido de alta frecuencia no muscular. Otros autores, por ejemplo, Venugopal et al. (2024) usaron 20–400 Hz en biceps dinámico y Senteler et al. (2022) emplearon 20–450 Hz para eliminar artefactos de movimiento [8]. Sin embargo, el ejercicio de levantamiento controlado de mancuernas se caracteriza porque:

- El espectro de potencia EMG muestra un desplazamiento hacia frecuencias más bajas lo que genera señales más estables y predecibles en comparación con movimientos rápidos, donde la fatiga acelera la pérdida de componentes altas [9].
- En contracciones isocinéticas repetidas, el espectro de potencia EMG se desplaza hacia frecuencias medias 50-150 Hz, con reducciones significativas en bandas altas

debido a la fatiga, concentrando la información neuromuscular relevante en el rango medio del espectro [10].

- Los artefactos de movimiento en ejercicios dinámicos controlados de baja velocidad ocurren principalmente en frecuencias menores a 20 Hz [5].

Por tanto, limitar el rango a 150 Hz permite optimizar la relación señal/ruido sin pérdida significativa de información útil, y se alinea con el objetivo del estudio de analizar con precisión la activación muscular durante un ejercicio controlado.

Envolvente Hilbertiana: La transformada de Hilbert se emplea habitualmente para calcular la envolvente de señales EMG, pues genera la señal analítica compleja cuya magnitud sigue los picos de amplitud del EMG como se ilustra en la Figura 6[11]. En la práctica, se toma la transformada analítica de la señal filtrada y luego su valor absoluto para obtener la envolvente instantánea.

Suavizado de la envolvente: Finalmente, la envolvente absoluta se suaviza mediante una ventana móvil de 50 muestras (25 ms) con el propósito de eliminar fluctuaciones rápidas residuales y obtener el sobre nivel de la actividad.

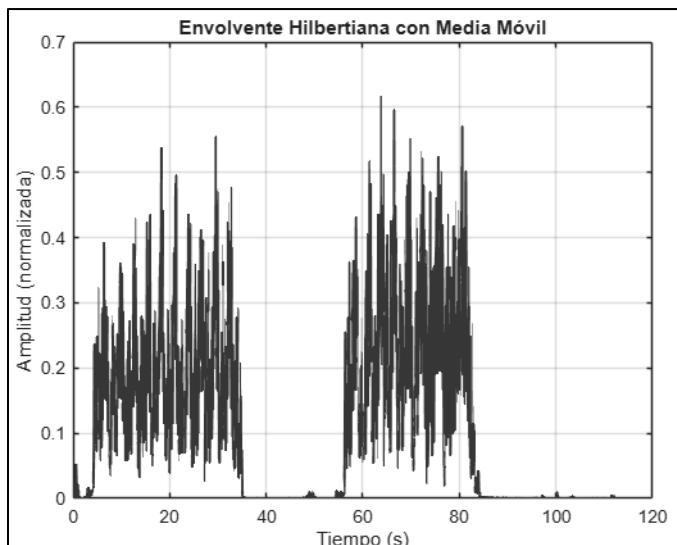


Fig 6: Envolvente de Hilbert suavizada de una señal electromiografía durante el levantamiento controlado de mancuernas.

Wu et al. (2021) destacan que las señales electromiográficas (EMG) son susceptibles al ruido de línea a 60 Hz, recomendando el uso de filtros notch para su supresión [4]. Boyer et al. (2023) revisan técnicas de filtrado para señales EMG, sugiriendo frecuencias de corte superiores a 20 Hz en filtros pasa-banda para eliminar artefactos de movimiento, y analizan los efectos de los filtros notch y pasa-banda en la calidad de la señal [5]. La transformada de Hilbert se utiliza ampliamente para extraer la envolvente de la señal EMG, permitiendo resaltar la amplitud instantánea [3]. Asimismo, el empleo de un filtro Butterworth de sexto orden, con respuesta plana en la banda pasante, está respaldado por literatura especializada en procesamiento de EMG [3] [5]. Estos hallazgos confirman que la cadena de preprocessamiento implementada es adecuada para discriminar de manera efectiva los períodos de reposo frente a contracción muscular en señales EMG del bíceps.

Procesamiento de los datos

El procesado de las señales se centra en la extracción de características discriminativas en el dominio temporal a partir de la envolvente de Hilbert suavizada, con el objetivo de capturar dinámicas electrofisiológicas asociadas a estados musculares de reposo y contracción durante ejercicios isométricos controlados, tal como indican otros estudios esta metodología de extracción de características aprovecha los parámetros de Hjorth para analizar señales, y transformarlas en un conjunto de descriptores cuantitativos que faciliten la clasificación automática, evitando de este modo el uso de métodos espectrales significativamente más cotosos a nivel computacional, priorizando así la eficiencia para potenciales aplicaciones en tiempo real, como monitoreo rehabilitativo o control de prótesis [12][13].

A continuación, en la tabla 6 se presentan las diferentes subetapas de procesado, y posteriormente se explica de manera clara por qué se eligió cada una, teniendo en cuenta aspectos relacionados con el funcionamiento

fisiológico y consideraciones prácticas ligadas a los algoritmos empleados.

#	Subetapa	Descripción
1	Segmentación en ventanas deslizantes	Divide la señal EMG en fragmentos para capturar dinámicas.
2	Cálculo de los parámetros de Hjorth	Extrae actividad, movilidad y complejidad de la señal.
3	Normalización de características	Estandariza datos para comparabilidad entre sujetos.
4	Etiquetado automático de estados musculares	Asigna categorías (reposo/contracción) a las ventanas.
5	Preparación para clasificación	Organiza características para algoritmos de machine Learning y Deep Learning.

Tabla 6: Pasos de Procesamiento de las envolventes obtenidas.

Justificación de las etapas de procesado

Segmentación en Ventanas Deslizantes: La envolvente suavizada fue segmentada mediante una ventana rectangular deslizante de 500 muestras, lo que equivale a 250 ms bajo una frecuencia de muestreo de 2000 Hz. Este tamaño de ventana permite capturar las rápidas variaciones de la activación muscular sin perder información temporal relevante, adicionalmente también se implementó un solapamiento del 90%, para equilibrar resolución temporal y robustez en la estimación de las características estadísticas de la forma de onda[14], el uso de ventanas rectangulares en la segmentación está fundamentado ya que el análisis se realiza sobre el dominio temporal, y esta forma no se afecta la amplitud de la señal durante la iteración del arreglo vectorial que contiene la información de la actividad muscular.

Este tipo de ventana tiene la ventaja de ser simple y eficiente, particularmente cuando se

trata de obtener una representación clara de la señal en una escala temporal definida y la longitud de ventana se alinea con escalas fisiológicas de reclutamiento de unidades motoras durante contracciones isométricas, permitiendo capturar transiciones entre reposo y activación sin una variabilidad excesiva intra-ventana [15].

Algunos estudios recientes justifican ventanas de 200 a 300 milisegundos en EMG para clasificación de gestos musculares, ya que minimizan el impacto del ruido estocástico y mejoran la precisión en entornos dinámicos, con incrementos reportados del 5-10% en métricas de separación de clases [16] mientras que el solapamiento parcial reduce artefactos de borde y asegura continuidad en análisis secuenciales, como se demuestra en enfoques de predicción de movimiento en tiempo real [17].

Cálculo de los Parámetros de Hjorth: Para cada ventana, se extraen los parámetros de Hjorth (actividad, movilidad y complejidad) directamente de la envolvente suavizada, como se muestra en la figura 7, que ilustra su evolución temporal.

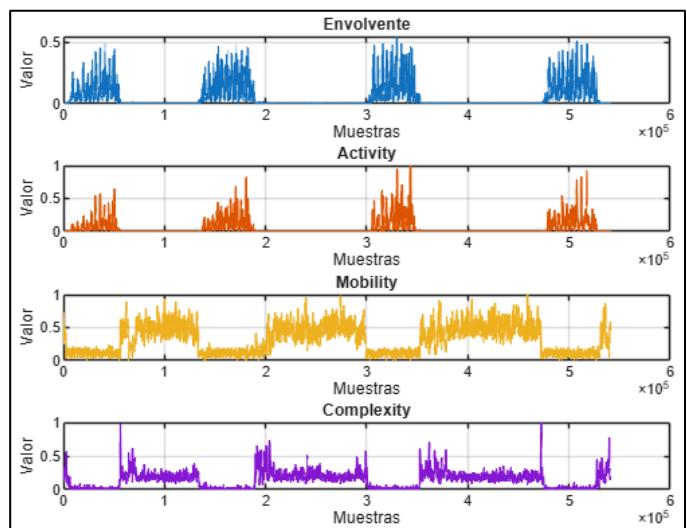


Fig. 7 Evolución temporal de la envolvente, actividad, movilidad y complejidad a lo largo de 5×10^5 muestras.

Estos descriptores temporales correlacionan con propiedades espectrales implícitas como la potencia, frecuencia media y ancho de banda, sin requerir transformadas

complejas, ofreciendo un costo computacional de $O(N)$ por ventana, lo cual es ideal para sistemas embebidos [18].

La actividad (A) se define como la varianza de la señal en la ventana, proporcionando una medida de la amplitud de la señal, que está relacionada con el nivel de activación muscular [19], elevándose significativamente durante contracciones debido a una mayor amplitud electrofisiológica:

$$A = \frac{1}{N} \sum_{t=1}^N (x(t) - \mu)^2 = Var(x(t))$$

La movilidad (M) mide la velocidad media de cambio, calculada como la raíz cuadrada de la relación entre la varianza de la primera derivada y la varianza de la señal:

$$M = \sqrt{\frac{Var(\dot{x}(t))}{Var(x(t))}} \therefore \dot{x}(t) \approx \frac{x(t+1) - x(t)}{\Delta t}$$

En este contexto la movilidad puede capturar información sobre desplazamientos frecuenciales durante la transición estados musculares de reposo a contracción [13].

Finalmente, la complejidad (C) evalúa la irregularidad de la forma de onda respecto a una sinusoidal pura y se define como la raíz de la varianza de la segunda derivada entre la varianza de la primera derivada, sobre la movilidad de la señal.

$$C = \frac{\sqrt{\frac{Var(\ddot{x}(t))}{Var(\dot{x}(t))}}}{M}$$

Para C, los valores cercanos a 1 indican patrones simples asociados al estado de reposo, mientras que variaciones mayores reflejan complejidad generan valores más cercanos a cero tal como se observa en la figura 7.

Para optimizar los tiempos de procesamiento las derivadas se aproximan mediante métodos numéricos empleando la operación de diferencias finitas con la función `diff` en MATLAB, estas se ajustaron al intervalo temporal $dt = 1/Fs$, para preservar precisión en señales muestradas y también se ignoró el sesgo marginal en varianzas por reducción de longitud.

Estos parámetros fueron elegidos por su robustez frente a ruido y correlación con métricas clínicas, superando features tradicionales como RMS en escenarios de bajo SNR, con precisiones reportadas >95% en clasificación de gestos [20].

Normalización de Características: Una vez calculados los parámetros de Hjorth, Cada registro se normalizó mediante min-max scaling individual, de modo que los valores oscilaron entre 0 y 1, reduciendo la variabilidad intersujeto debida a diferencias en amplitud basal o impedancia de electrodos.

Esta normalización mejora la generalización en modelos de machine learning, reduciendo sesgos y facilitando convergencia en algoritmos como SVM o redes neuronales, como se evidencia en revisiones sistemáticas donde incrementa la precisión en 3-5% para datasets heterogéneos [21].

La salida por ventana se estructura como un vector [A, M, C], conformando el conjunto de características para análisis subsiguiente.

Etiquetado Automático de Estados: para etiquetar las características extraídas se implementó el algoritmo K-means con $K = 2$, tal como podemos observar en la figura 8, donde se muestra la asignación de datos por clústeres, esta técnica permite capturar patrones electrofisiológicos basados en umbrales implícitos de la actividad muscular, mientras que una distinción entre clústeres con un coeficiente de silueta de 0.6648, indica una separación moderada pero consistente confirmando la existencia de estructuras discriminables.

Aunque este método es efectivo, para casos diferentes a la actividad muscular isométrica, podría ser recomendable adaptar el ancho de ventana.

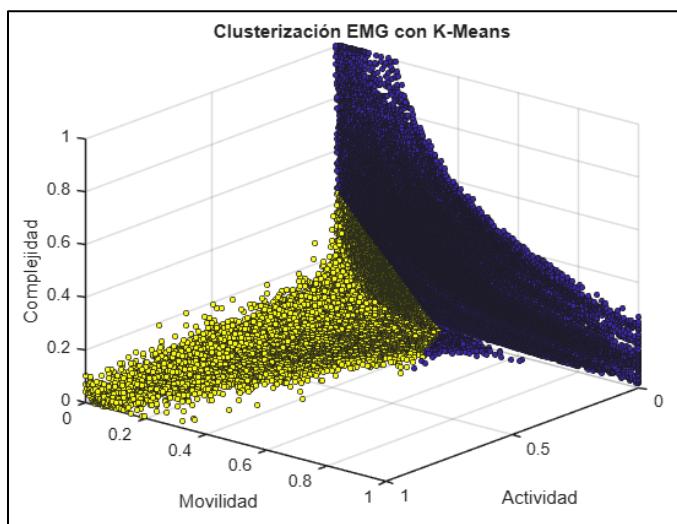


Fig. 8 Clusterización K-means de características EMG en espacio 3D.

Dado que los datos se agrupan inicialmente de manera arbitraria se han calculado la media de los parámetros por clúster (vea la tabla 7) reflejando diferencias en actividad y complejidad.

Clúster	Estado	Actividad	Movilidad	Complejidad
1	Reposo	0.0001	0.4037	0.2562
2	Contracción	0.1051	0.1303	0.0367

Tabla 7: Medias de las características por clúster.

Preparación para la clasificación: El conjunto de características etiquetadas fue exportada al archivo emg_features_clustered.csv, la tabla incluye 472,002 muestras con variables las Activity, Mobility, Complexity y Cluster.

En esta etapa se enfatizó la escalabilidad, con características Hjorth demostrando superioridad en clasificación superando métodos espectrales en tiempo de inferencia (<0.05 s) para aplicaciones clínicas [20].

Optimización de la base de datos

Para mejorar la eficiencia computacional se realizó una optimización de la base de datos. El proceso transformó el archivo inicial

emg_features_clustered.csv. reducido emg_features_optimized.csv mediante ajustes en los tipos de datos, preservando la integridad de las características para entrenamientos posteriores.

Inicialmente la tabla de características se carga desde emg_features_clustered.csv utilizando la función readtable de MATLAB, resultando en una estructura de $472,001 \times 4$ con variables Activity, Mobility, Complexity y Clúster, almacenadas inicialmente como double (64-bit), ocupando 15.1 MB en memoria, las estadísticas iniciales se presentan en la Tabla 8, mostrando un desbalance hacia el clúster 2 que contiene un 60% del total de las muestras.

Variable	Min	Median	Max	Mean	Std
Activity	0	0.0016	1	0.0633	0.1189
Mobility	0	0.1695	1	0.2392	0.1737
Complexity	0	0.0470	1	0.1241	0.1458
Clúster	1	2	2	1.6018	0.4895

Tabla 8: Estadísticas Iniciales de la Tabla con variables tipo double, tamaño total: $472,001 \times 4$, 15.1 MB.

Para optimizar el almacenamiento, las variables Activity, Mobility y Complexity se convirtieron de tipo double a half (16-bit float), reduciendo el tamaño por un factor de compresión ~4.6x, mientras que la variable Cluster se transformó a tipo booleano eliminando redundancia, ocupando así un tamaño total de 3.3 MB en memoria lo que supone una reducción de ~78.15%. Las estadísticas post optimización se muestran en la Tabla 9.

Variable	Min	Median	Max	Mean	Std
Activity	0	0.0016	1	0.0633	0.1189
Mobility	0	0.1695	1	0.2392	0.1737
Complexity	0	0.0470	1	0.1241	0.1458
Cluster	0	1	1	0.6018	N/A

Tabla 8: Estadísticas post optimización de la Tabla con variables tipo double, tamaño total: $472,001 \times 4$, 3.3 MB.

Finalmente se verificó la integridad de los datos comparando las estadísticas pre y post

optimización, asegurando que mean y std se mantengan dentro de una tolerancia mínima <0.01% de variación.

Modelos de clasificación

En esta sección se presentan los modelos de aprendizaje automático y profundo evaluados con el objetivo de comparar sus desempeños en la clasificación de señales EMG, se implementaron y compararon distintos algoritmos listados en la tabla 9 con el fin de contrastar precisión, eficiencia computacional y escalabilidad.

N	Modelo
1	Decision Tree (DT)
2	Random Forest (RF)
3	K-Nearest Neighbors (KNN)
4	Support Vector Machine (SVM)
5	Convolutional Neural Network (CNN)
6	Long Short-Term Memory (LSTM)
7	CNN-LSTM

Tabla 9. Modelos de clasificación evaluados

A continuación, se presenta un resumen de los modelos considerados.

Modelos clásicos

Decision Tree (DT): es un modelo supervisado que organiza decisiones como una estructura de árbol, cada nodo interno realiza una prueba sobre una característica, cada rama representa una posible salida de la prueba, y cada hoja corresponde a una etiqueta de clase, bajo esta representación el problema de clasificación se convierte en particiones sucesivas del espacio de características [22].

Random Forest: es un método *ensemble* que construye múltiples árboles de decisión independiente, cada uno entrenado sobre muestras distintas del conjunto de datos y luego combina sus predicciones por ejemplo mediante votación, con el fin de reducir la varianza del

modelo y mejorar la generalización respecto a un solo árbol [23]

KNN: es un clasificador supervisado no paramétrico que decide la clase de una muestra nueva considerando las clases de sus *k* vecinos más cercanos en el espacio de características que no requiere entrenamiento explícito más allá del almacenamiento de los ejemplos de entrenamiento [24].

SVM: es un modelo supervisado que busca determinar un hiperplano óptimo que separa las clases maximizando el margen entre los vectores de soporte. En casos no lineales, utiliza funciones kernel para mapear los datos a un espacio donde la separación lineal es factible [25].

Modelos profundos

CNN: son arquitecturas profundas diseñadas para aprender características locales mediante capas de convolución y pooling compartiendo pesos. En el contexto de series temporales las CNN pueden extraer automáticamente representaciones jerárquicas directamente desde los datos sin necesidad de ingeniería manual de características [26].

LSTM: es una variante de red neuronal recurrente (RNN) diseñada para aprender dependencias de largo plazo en secuencias, superando los problemas de desvanecimiento del gradiente mediante puertas que regulan el almacenamiento, olvido y actualización de la información [27].

CNN-LSTM: es una arquitectura híbrida que integra la capacidad de las CNN para extraer características espaciales locales y la de las LSTM para capturar las dependencias temporales en una secuencia. Combinando ambos enfoques, se pretende aprovechar lo mejor de cada modelo para datos secuenciales que contienen estructura espacial y temporal [28].

Aunque el preprocesamiento de las señales se realizó en MATLAB, extrayendo características

de envolvente de la señal y parámetros de Hjorth, el entrenamiento y evaluación de los modelos se implementó en Python utilizando la plataforma Google Colab,

Métricas de evaluación

Los modelos se evaluaron utilizando métricas estándar de clasificación binaria (Tabla 10): exactitud (accuracy), precisión (precision), exhaustividad (recall) y puntaje F1, junto con los promedios macro y ponderado (weighted avg). Además, se calculó el tiempo de inferencia medio por muestra como indicador del costo computacional. Para un análisis más detallado del desempeño, las métricas se reportaron por clase (0 y 1), lo que permitió evaluar tanto el equilibrio del modelo entre clases como su rendimiento global considerando el posible desbalance en los datos.

Métrica	Definición
Tiempo de inferencia (CPU/GPU)	Promedio del tiempo requerido para procesar una muestra durante la fase de inferencia.

Tabla 10. Métricas de desempeño utilizadas para la evaluación de los modelos de clasificación.

Todos los modelos fueron entrenados con el 80 % de los datos y validado con el 20 % restante, garantizando una partición reproducible de las señales electromiográficas procesadas. La base de datos final incluyó los metadatos de 472 001 segmentos, cada uno caracterizado por las variables Activity, Mobility y Complexity, y etiquetado según el estado muscular mediante la etiqueta Clúster (0 reposo; 1 actividad).

Resultados iniciales

Decision Tree: El modelo de árbol de decisión se empleó como punto de partida para la evaluación comparativa de clasificadores supervisados, se presentan los resultados obtenidos mediante el reporte de clasificación de Scikit-learn (tabla 11), así como la matriz de confusión (figura 9) y coto computacional (tabla 12).

Clase	Precision	Recall	F1-Score	Support
0	0.9975	0.9979	0.9977	150 475
1	0.9986	0.9984	0.9985	227 126
Accuracy			0.9982	377 601
Macro avg	0.9981	0.9981	0.9981	377 601
Weighted avg	0.9982	0.9982	0.9982	377 601

Tabla 11. Resultados del modelo Decision Tree.

Estos valores evidencian una capacidad de generalización sobresaliente, además la matriz de confusión confirmó una mínima tasa de falsos positivos y falsos negativos, indicando una discriminación robusta entre los estados musculares de reposo y actividad.

Métrica	Definición
Exactitud (Accuracy)	Proporción de predicciones correctas respecto al total de muestras evaluadas.
Precisión (Precision)	Proporción de verdaderos positivos entre todas las predicciones positivas realizadas por el modelo.
Exhaustividad (Recall)	Proporción de verdaderos positivos correctamente identificados entre todas las muestras realmente positivas.
Puntaje F1	Media armónica entre precisión y exhaustividad, que representa el equilibrio entre ambas métricas.
Macro avg	Promedio aritmético de la métrica calculada por clase, trata todas las clases por igual.
Weighted avg	Promedio ponderado de la métrica, donde cada clase contribuye según su soporte.

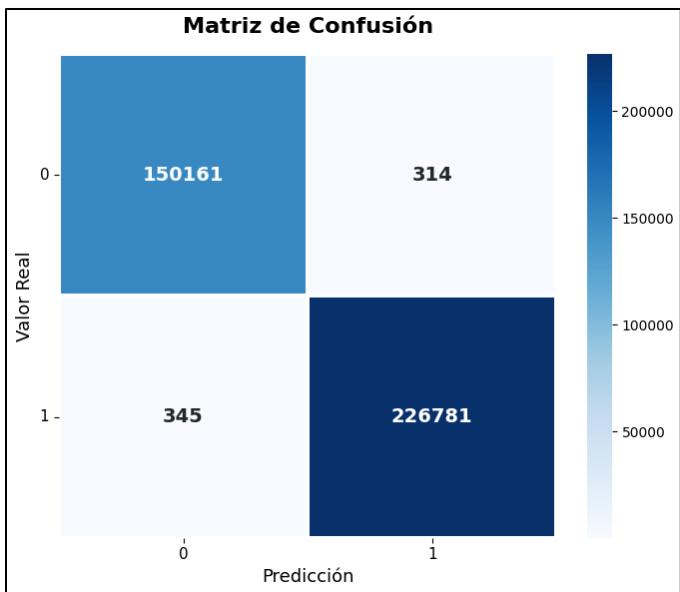


Fig. 9 Matriz de confusión Decision Tree.

Adicionalmente, se estimó el costo computacional del modelo para valorar su aplicabilidad en entornos de tiempo real.

Parámetro	Valor medido
Tiempo medio de inferencia	0.020 s \pm 0.004 s
Uso de RAM	\approx 342 MB
Espacio en disco del modelo exportado	0.027 MB

Tabla 12. Análisis de rendimiento Decision Tree, basado en 1000 iteraciones.

Pruebas

El modelo de Decision Tree se evaluó generando predicciones crudas, mostrando alta exactitud general pero con una tasa de falsos negativos (FN) elevada durante los períodos de actividad muscular, posiblemente debido al desbalance de datos lo que resulta en una detección subóptima donde el modelo tiende a clasificar erróneamente muestras de contracción como reposo (véase la figura 10), especialmente en segmentos con variabilidad alta, con el fin de mitigar esto se propone un suavizado de probabilidad con umbral añadido, obteniendo los resultados de la figura 11.

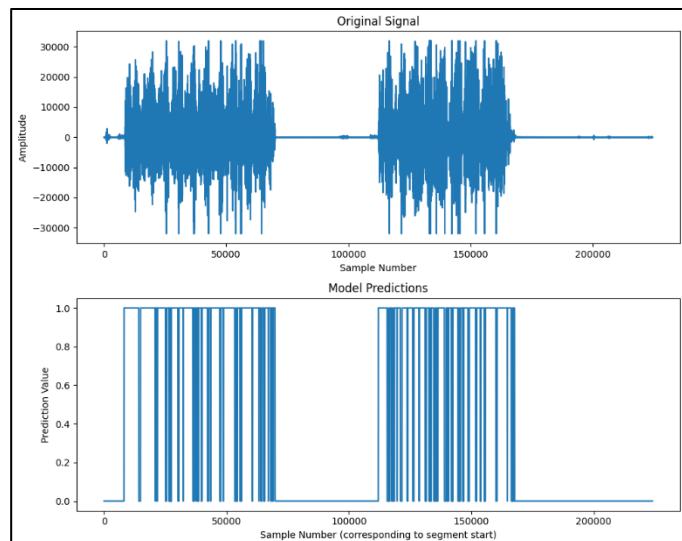


Fig. 10. Señal de entrada versus predicciones del modelo con alta tasa de FN.

El suavizado de probabilidad con umbral añadido consiste en aplicar una media móvil con ventana de 51 muestras (~ 25.5), seguida de un umbral de 0.1 para binarizar la actividad, esta tarea se realizó sobre el vector que contenía las predicciones del modelo. La ventana de 51 considera la influencia de predicciones vecinas, promediando fluctuaciones locales, mientras que el umbral de 0.1 permite eliminar artefactos del vector de predicciones.

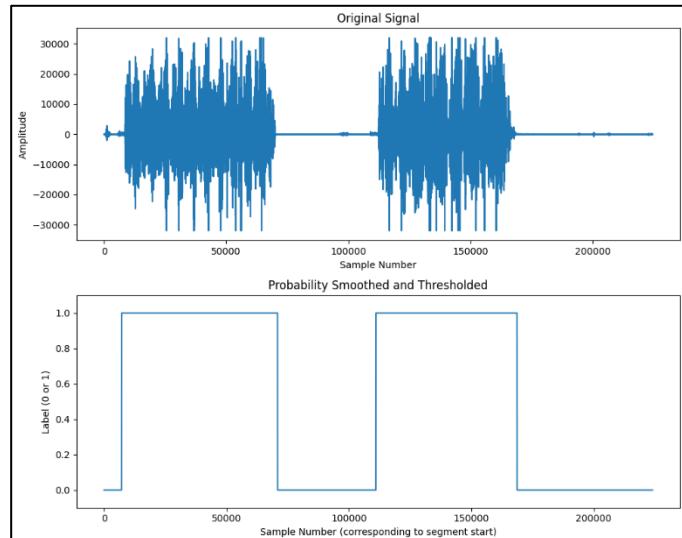


Fig. 11. Señal de entrada versus vector de predicciones después del postprocesado.

Adicionalmente, se introdujeron pruebas con tríceps braquial, un músculo antagonista al bíceps involucrado en extensiones, utilizando la misma metodología obteniendo resultados preliminares similares (figura 12) lo que parece

ser un buen indicio de la robustez del modelo en conjunto con la etapa de postprocesado.

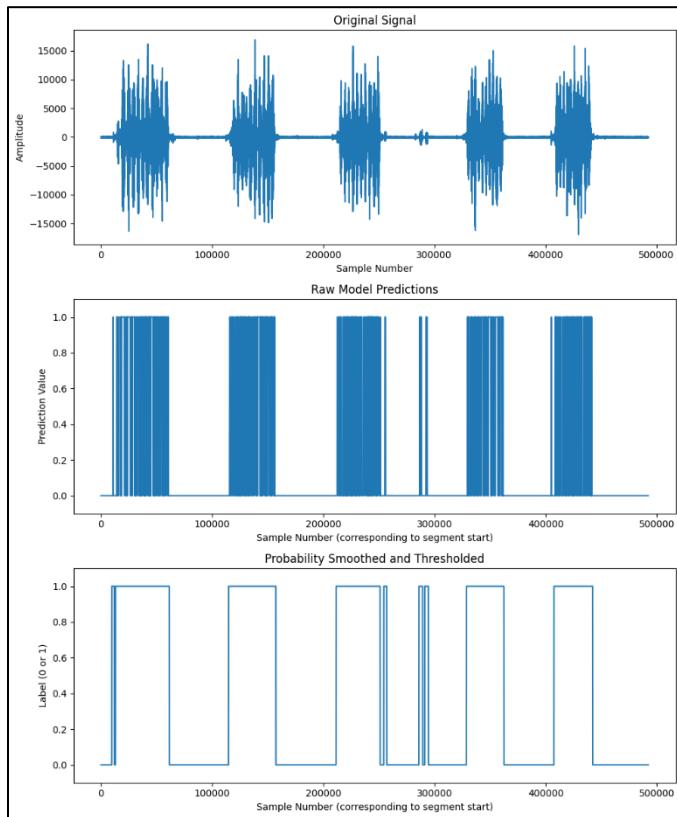


Fig. 12. EMG bíceps braquial de entrada versus vector de predicciones aplicando la metodología de postprocesado propuesto para mitigar la tasa de FN.

Los resultados obtenidos con el modelo de Decision Tree, tanto para el bíceps como para el tríceps, destacan su robustez inicial y la efectividad del posprocesamiento con suavizado de probabilidad y umbralización en la reducción de falsos negativos, mejorando la detección de estados musculares en entornos desbalanceados. La generalización a músculos antagonistas como el tríceps sugiere un potencial amplio para aplicaciones multimúsculo, aunque ajustes finos en los parámetros podrían optimizar aún más el desempeño.

Random Forest: El modelo se implementó para mejorar la generalización observada en el Decision Tree. A continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Scikit-learn (Tabla 13), la matriz de confusión (Figura 13) y el costo computacional (Tabla 14).

Clase	Precision	Recall	F1-Score	Support
0	0.9994	0.9994	0.9994	37671
1	0.9996	0.9996	0.9996	56730
Accuracy			0.9995	94401
Macro avg	0.9995	0.9995	0.9995	94401
Weighted avg	0.9995	0.9995	0.9995	94401

Tabla 13. Resultados del modelo Random Forest.

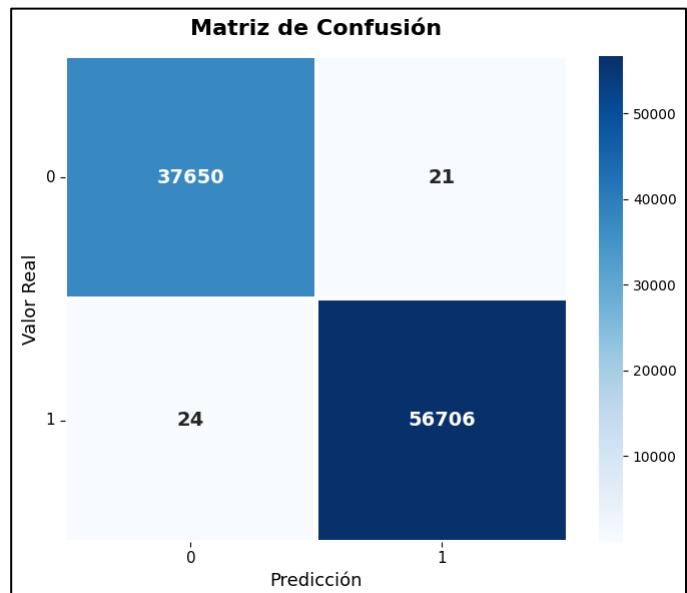


Fig. 13. Matriz de confusión Random Forest.

Estos valores evidencian una capacidad de generalización superior, con una discriminación robusta sin embargo al evaluar el rendimiento se evidencia un aumento significativo en los tiempos de inferencia del modelo, así como de los recursos computacionales necesarios.

Parámetro	Valor medido
Tiempo medio de inferencia	0.433779 ± 0.067348 s
Uso de RAM	405.164062 MB
Espacio en disco del modelo exportado	12.059125 MB

Tabla 14. Análisis de rendimiento Random Forest, basado en 1000 iteraciones.

Pruebas

Aunque este modelo combina múltiples árboles de decisión, entrenados con submuestras del dataset mediante muestreo aleatorio con reemplazo, durante las pruebas se

observó un aumento de los FN (figura 14), esto es posiblemente atribuible al sesgo hacia la clase mayoritaria del dataset desbalanceado.

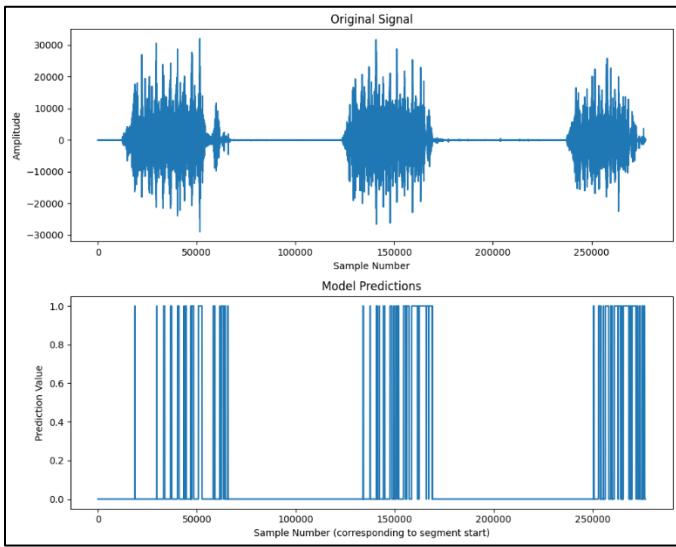


Fig. 14. Señal de entrada versus predicciones del modelo con alta tasa de FN.

Este problema que se mitiga gracias al postprocesamiento igual que en el caso del Decision Tree, pero ajustando los parámetros de ancho y umbral de la función de suavizado (figura 15), sin embargo, considerando el aumento de costo computacional en comparación con el Decisión Tree esta no parece ser una alternativa tan prometedora para sistemas embebidos.

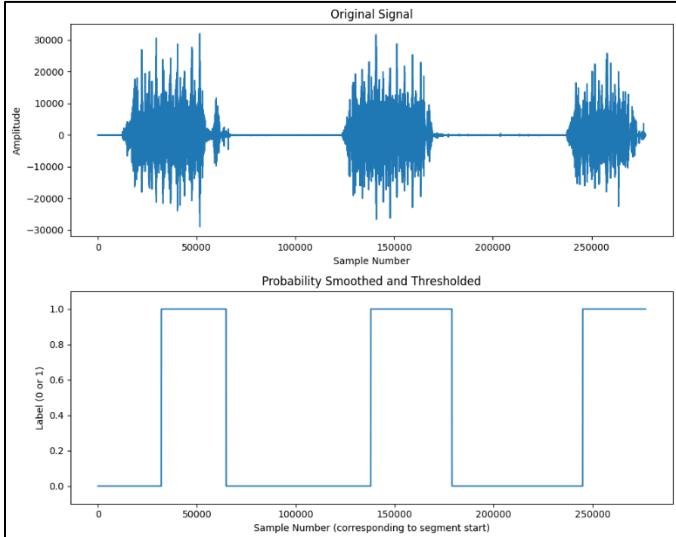


Fig. 15. Señal de entrada versus vector de predicciones después del postprocesado.

Por otro lado, las pruebas extendidas al tríceps braquial refuerzan esta observación, mostrando un comportamiento similar (Figura 16) lo que abre la posibilidad de optimizar modelo

para señales más estables, mientras que para el bíceps, el Decision Tree sigue siendo preferible.

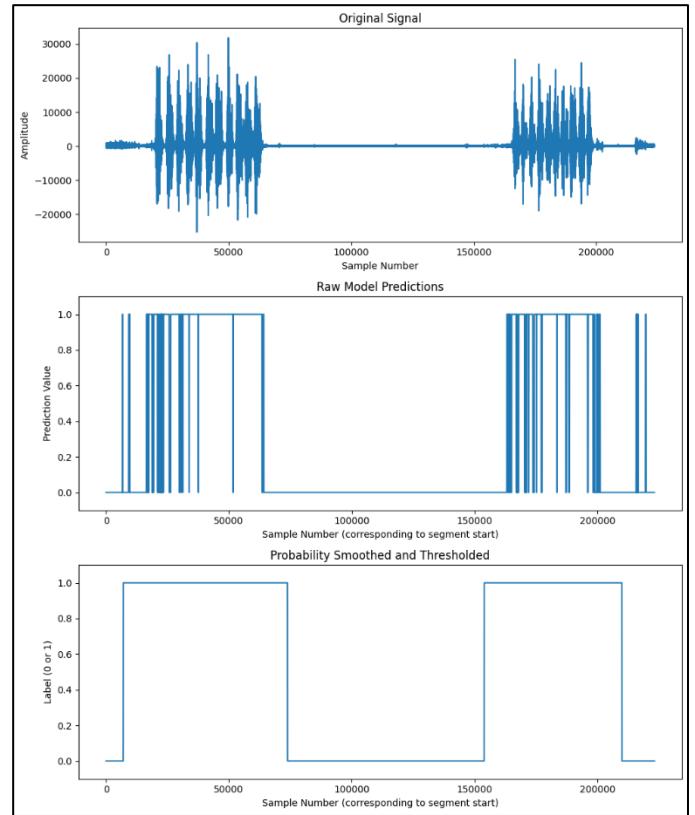


Fig. 16. EMG bíceps braquial de entrada versus vector de predicciones aplicando la metodología de postprocesado propuesto para mitigar la tasa de FN.

K-Nearest Neighbors: este es un clasificador supervisado no paramétrico que decide la clase de una muestra nueva considerando las clases de sus k vecinos más cercanos en el espacio de características, a continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Scikit-learn (Tabla 15), la matriz de confusión (Figura 17) y el análisis de costo computacional (Tabla 16).

Clase	Precision	Recall	F1-Score	Support
0	0.9996	0.9994	0.9995	94,070
1	0.9996	0.9997	0.9997	141,931
Accuracy			0.9996	236,001
Macro avg	0.9996	0.9996	0.9996	236,001
Weighted avg	0.9996	0.9996	0.9996	236,001

Tabla 15. Resultados del modelo K-Nearest Neighbors (KNN).

Los resultados preliminares muestran un rendimiento superior tanto al Decision Tree como al Random Forest en términos de precisión global y equilibrio entre clases.

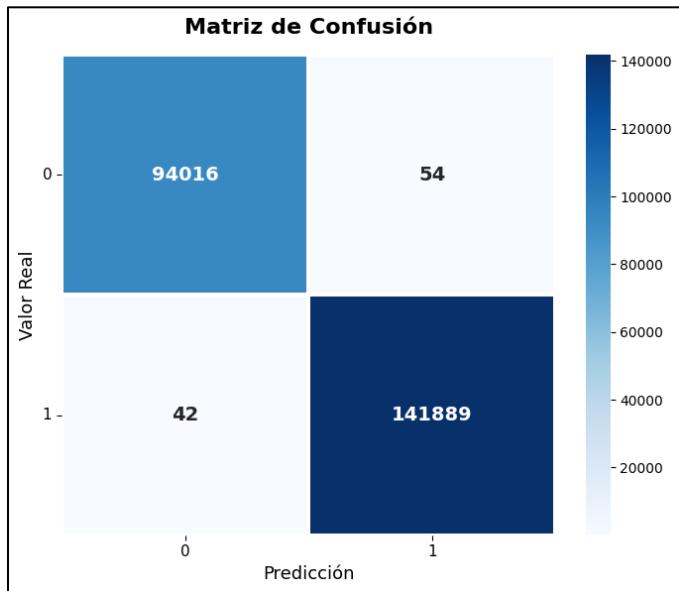


Fig. 17. Matriz de confusión K-Nearest Neighbors.

El análisis de la tasa de error en función del número de vecinos (k) reveló que valores bajos de k (1-5) ofrecen el mejor desempeño, con tasas de error mínimas (figura 18), confirmando la efectividad del método en este contexto de clasificación binaria



Fig. 18. Tasa de error vs número de vecinos.

Adicionalmente, también se estimó el costo computacional del modelo para valorar su aplicabilidad en entornos de tiempo real.

Parámetro	Valor medido
Tiempo medio de inferencia	3.018979 s ± 0.395356 s
Uso de RAM	356.417969 MB
Espacio en disco del modelo exportado	9.628885 MB

Tabla 16. Análisis de rendimiento K-Nearest Neighbors, basado en 1000 iteraciones.

Con estos resultados, se observa un aumento significativo en los tiempos de inferencia, igualmente se procede a evaluar el modelo.

Pruebas

Se observa un patrón similar a modelos anteriores con una alta exactitud general, pero con una tasa de falsos negativos (FN) durante períodos de actividad muscular, mostrando que el modelo tiende a clasificar erróneamente muestras de contracción como reposo, especialmente en segmentos con alta variabilidad (véase Figura 17).

En este punto se considera la posibilidad de revisar manualmente las etiquetas de los 472,002 registros para garantizar su fiabilidad, además de técnicas como un overlap más agresivo al conjunto de actividad para corregir el desbalance del dataset a costa de la variabilidad de los datos y también descartar la variable de movilidad que parece ser la que menos información útil aporta, por ahora esto es solo una idea, sin embargo, se modificará el conjunto de datos para modelos profundos.

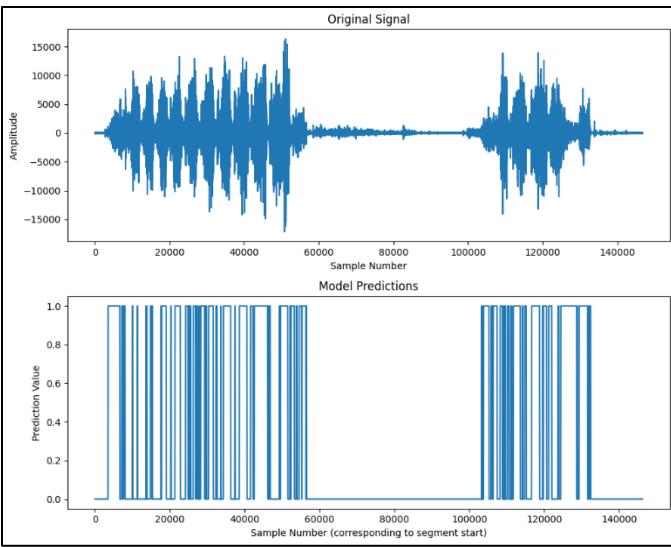


Fig. 17. Señal de entrada versus predicciones crudas del modelo KNN con tasa de FN elevada.

El suavizado de probabilidad aplicado sobre las probabilidades de predicción reduce significativamente los FN aislados mejorando la continuidad temporal de las detecciones (Figura 18).

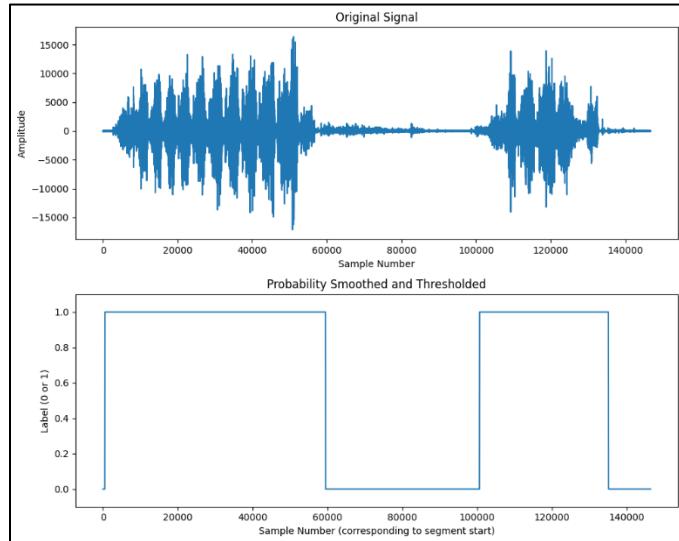


Fig. 18. Señal de entrada versus vector de predicciones KNN después del postprocesamiento.

se introdujeron pruebas con el tríceps braquial, los resultados preliminares muestran una generalización efectiva del modelo KNN a este músculo diferente, con una tasa de FN similar a señales del bíceps durante contracciones (Figura 19).

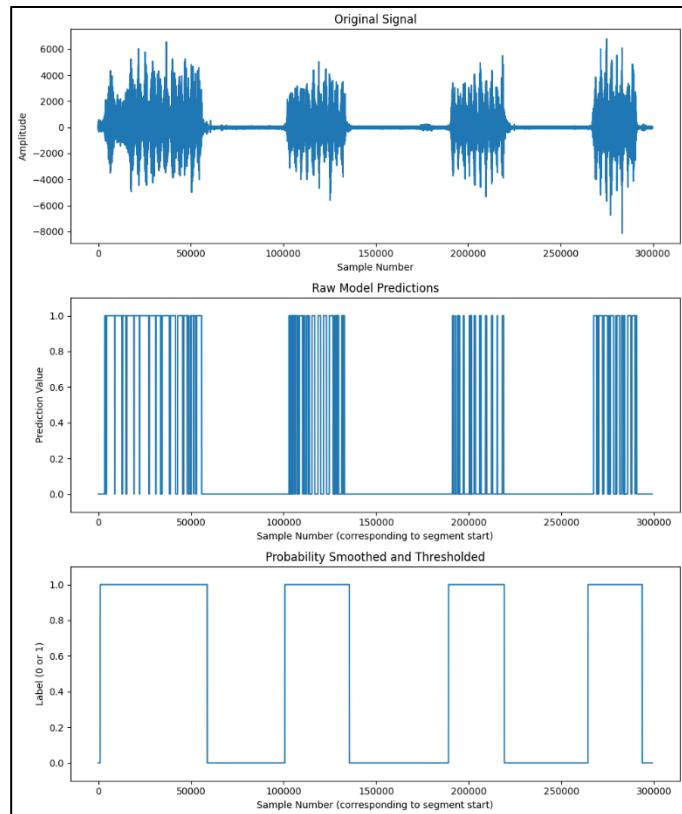


Fig. 19. EMG tríceps braquial procesado con KNN y postprocesamiento, mostrando robustez en señales de extensión muscular.

La generalización exitosa a músculos antagonistas sugiere que las características de Hjorth, combinadas con la aproximación basada en distancias locales del KNN, capturan patrones electrofisiológicos fundamentales independientemente del tipo muscular específico.

Support Vector Machines: este es un algoritmo supervisado que busca determinar un hiperplano óptimo que maximiza el margen entre las clases, utilizando vectores de soporte como puntos críticos para la separación, a continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Scikit-learn (Tabla 17), la matriz de confusión (Figura 19) y el análisis de costo computacional (Tabla 18).

Clase	Precision	Recall	F1-Score	Support
0	0.9999	0.9997	0.9998	56,414
1	0.9998	1.0000	0.9999	85,187
Accuracy			0.9999	141,601

Clase	Precision	Recall	F1-Score	Support
Macro avg	0.9999	0.9998	0.9999	141,601
Weighted avg	0.9999	0.9999	0.9999	141,601

Tabla 17. Resultados del modelo Support Vector Machines (SVM).

Este rendimiento anormalmente alto podría indicar que el modelo memorizado los datos de entrenamiento en lugar de aprender a generalizar, lo que lo haría incapaz de predecir o clasificar con precisión datos nuevos, lo que es un fenómeno común cuando el espacio de características está sobre-optimizado, mas adelante se evaluara el comportamiento del modelo tal cual se ha hecho con los anteriores realizando también pruebas extendidas al tríceps braquial.

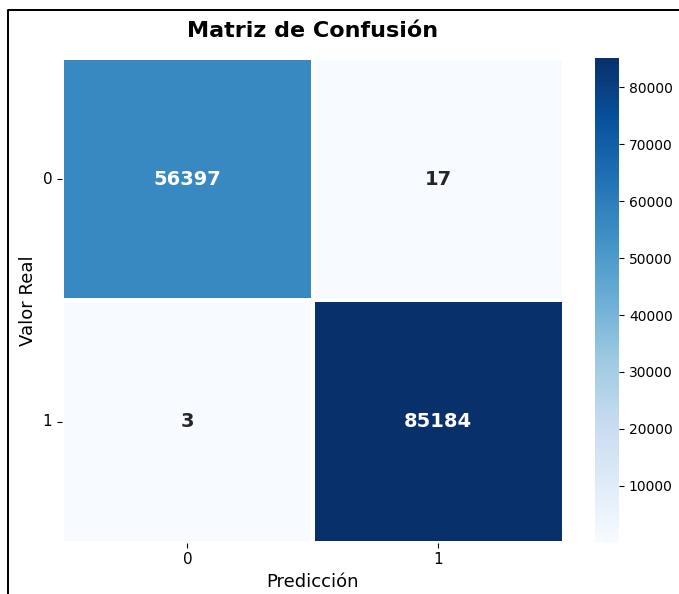


Fig. 19. Matriz de Confusión Support Vector Machines.

Al igual que se realizó anteriormente también se estimó el costo computacional del modelo para evaluar su aplicabilidad en entornos de tiempo real indicando que este modelo además de ser el computacionalmente mas costoso también muestra tiempos de inferencia muy superiores hasta los obtenidos hasta el momento, lo cual podría justificarse aludiendo a su mayor robustes, sin embargo, esta característica lo hace poco viable para su aplicación en sistemas embebidos.

Parámetro	Valor medido
Tiempo medio de inferencia	19.890542 ± 2.338123 s
Uso de RAM	637.207031 MB
Espacio en disco del modelo exportado	0.107649 MB

Tabla 18. Análisis de rendimiento Support Vector Machines, basado en 100 iteraciones.

Pruebas

Se observa un patrón similar a modelos anteriores con una alta exactitud general, pero con una tasa de falsos negativos durante periodos de actividad muscular, mostrando que el modelo tiende a clasificar erróneamente muestras de contracción como reposo, especialmente en segmentos con alta variabilidad como se observa en la Figura 20.

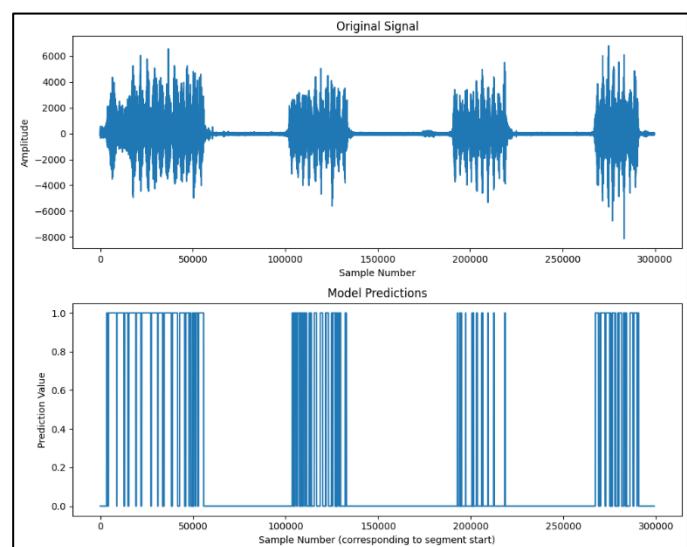


Fig. 20. Señal de entrada versus predicciones.

El suavizado de probabilidad aplicado sobre las probabilidades de predicción reduce significativamente los FN aislados, mejorando la continuidad temporal de las detecciones (Figura 21). Este método considerar la influencia de predicciones vecinas eliminando artefactos del vector de predicciones lo cual ha resultado ser una estrategia exitosa hasta ahora.

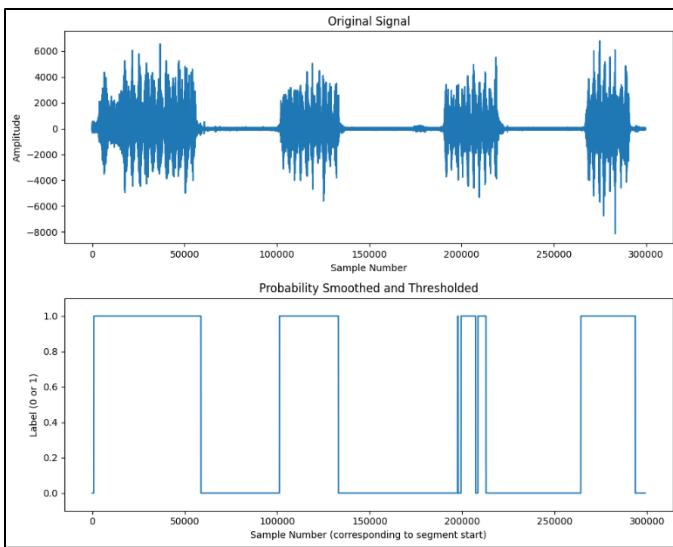


Fig. 21. Señal de entrada versus vector de predicciones SVM después del postprocesamiento.

Adicionalmente, se introdujeron pruebas con el tríceps braquial, los resultados preliminares muestran una generalización efectiva, con una tasa de FN similar a señales del bíceps durante contracciones (Figura 22).

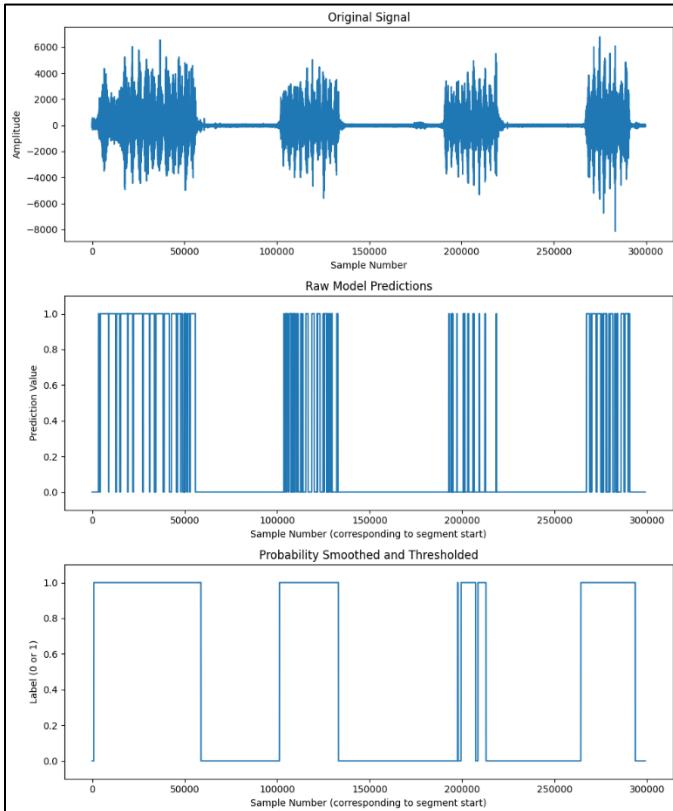


Fig. 22. EMG tríceps braquial procesado con SVM y postprocesamiento, mostrando robustez en señales de extensión muscular.

Resumen de las pruebas ML

Esta sección presenta un pequeño resumen de análisis comparativo de los modelos de aprendizaje automático, aquí se detallan los puntos clave:

- Generalización:** Todos los modelos generalizan bien a tríceps, con altas tasas de falsos negativos principalmente observadas durante las pruebas de desempeño, estas fueron reducidas mediante técnicas de postprocesamiento que consistían en filtros de media móvil y umbralización.
- Limitaciones:** SVM y KNN enfrentan tiempos de inferencia altos (~19.89 s), mientras que Decision Tree y Random Forest son más viables para su aplicación en sistemas de baja potencia.
- Postprocesamiento:** El suavizado con media móvil y umbralización mejora la detección, especialmente en datos desbalanceados, sin embargo, se ha planteado la necesidad de reevaluar la estructura del dataset para la etapa posterior de evaluación (modelos de aprendizaje profundo).

Aunque los resultados son prometedores para aplicaciones clínicas y deportivas, se pueden optimizar el desbalanceo de clases, así como los hiperparámetros mejorar el desempeño más el desempeño, veamos la tabla 19 y 20 donde se presentan un resumen de las métricas de cada modelo.

Modelo	Precisión	Recall	F1-Score	Inferencia
Decision Tree	~0.98 / ~0.97	~0.97 / ~0.98	~0.975 / ~0.975	~0.01s
Random Forest	0.99 / 0.98	0.99 / 0.98	0.99 / 0.98	~0.0411s
KNN	~0.98 / ~0.97	~0.96 / ~0.99	~0.97 / ~0.98	~1-2 s
SVM	0.9999 / 0.9998	0.9997 / 1.0	0.9998 / 0.9999	~19.89 s

Tabla 19. Resumen de métricas ML (durante el entrenamiento).

Modelo	RAM	Tamaño	Viabilidad Tiempo Real
Decision Tree	~342 MB	~0.03 MB	Muy Alta
Random Forest	~405 MB	~12.1 MB	Alta
KNN	~356 MB	~10 MB	Baja
SVM	~637 MB	~0.1 MB	Muy Baja

Tabla 20. Resumen de costo computacional modelos ML (durante el entrenamiento).

Una vez evaluados los modelos de aprendizaje automático clásico, que demostraron altas precisiones ($>96\%$), se procede a explorar enfoques de aprendizaje profundo para capturar dependencias temporales y espaciales más complejas en las señales EMG.

Resultados iniciales

CNN: este es un clasificador profundo basado en convoluciones que extrae patrones locales de las entradas, diseñado para procesar datos estructurados como imágenes 1D derivadas de las características de Hjorth optimizadas (actividad y complejidad), a continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Keras (Tabla 21), la matriz de confusión (Figura 25) y el análisis de costo computacional (Tabla 22).

Clase	Precision	Recall	F1-Score	Support
0	0.9994	0.9998	0.9996	56,414
1	0.9999	0.9996	0.9997	85,187
Accuracy			0.9997	141,601
Macro Avg	0.9996	0.9997	0.9997	141,601
Weighted Avg	0.9997	0.9997	0.9997	141,601

Tabla 21. Resultados del modelo Convolutional Neural Network (CNN).

Los resultados preliminares muestran un rendimiento similar a modelos anteriores en términos de precisión global, sin embargo, en

etapas anteriores se observó un desempeño con una alta tasa de FN durante el periodo de implementación, en este caso se presenta una matriz de confusión que refleja este hecho incluso antes, pero no se puede inferir sobre el desempeño del modelo en este punto, como consecuencia se procede con las pruebas para evaluar el rendimiento.

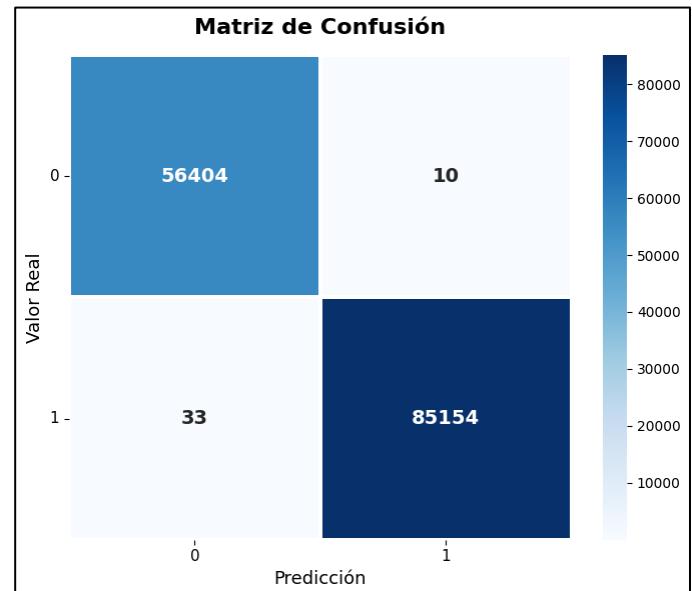


Fig. 25. Matriz de confusión Convolutional Neural Network

Adicionalmente, también se estimó el costo computacional del modelo para valorar su aplicabilidad en entornos de tiempo real.

Métrica	Valor
Tiempo de inferencia	0.074685 ± 0.020444 s
Uso de RAM	1661.480 MB
Espacio en disco del modelo	0.060722 MB

Tabla 22. Análisis de rendimiento Convolutional Neural Network, basado en 10 iteraciones.

Pruebas

El modelo se evaluó utilizando señales EMG cruda sobre las cuales se aplicó la misma cadena de preprocesamiento y extracción de características descrita en las fases previas (procesado)

Se observa un patrón similar a modelos anteriores con una alta exactitud general, pero al ser un modelo más robusto y complejo retorna un nivel de probabilidad, ya no se una alta tasa de falsos negativos durante periodos de actividad muscular, pero si una tendencia a un mayor nivel de duda frente al mismo escenario, mostrando que el modelo tiende a confundir muestras de contracción como reposo, vea la figura 25.

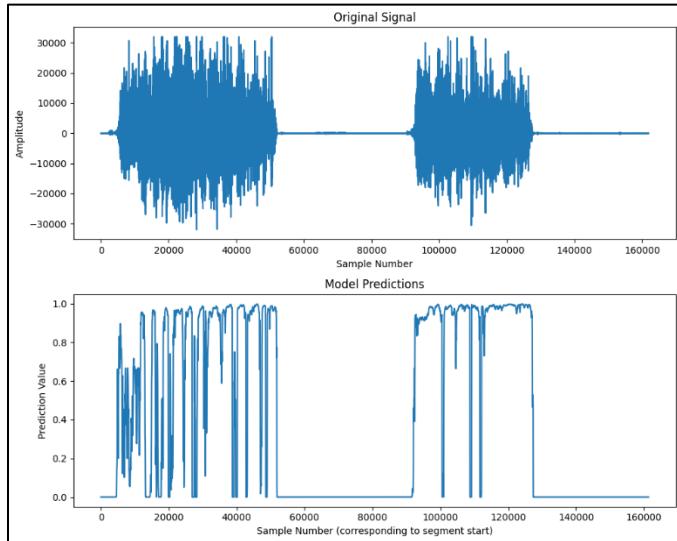


Fig. 25. Señal de entrada versus predicciones crudas del modelo CNN con tasa de confusión elevada.

El suavizado de probabilidad aplicado sobre las probabilidades de predicción reduce significativamente los casos de confusión aislados, mejorando la continuidad temporal de las detecciones (Figura 26).

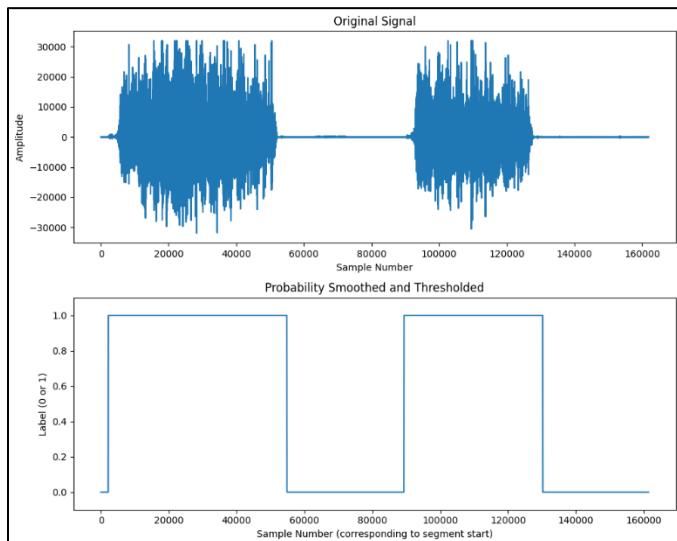


Fig. 26. Señal de entrada versus vector de predicciones CNN después del postprocesamiento.

Este método considera la influencia de predicciones vecinas mediante una media móvil con ventana de 150 muestras (~75 ms) y umbralización en 0.1, eliminando artefactos del vector de predicciones, lo cual ha resultado ser una estrategia exitosa hasta ahora.

Dado que los resultados preliminares de modelos clásicos con la misma metodología sugieren una generalización efectiva al tríceps braquial, con una tasa de FN similar a las señales del bíceps durante contracciones. Se espera un comportamiento análogo para el CNN (ejemplo en la figura 27), dada su capacidad para aprender patrones jerárquicos robustos a partir de las características de Hjorth.

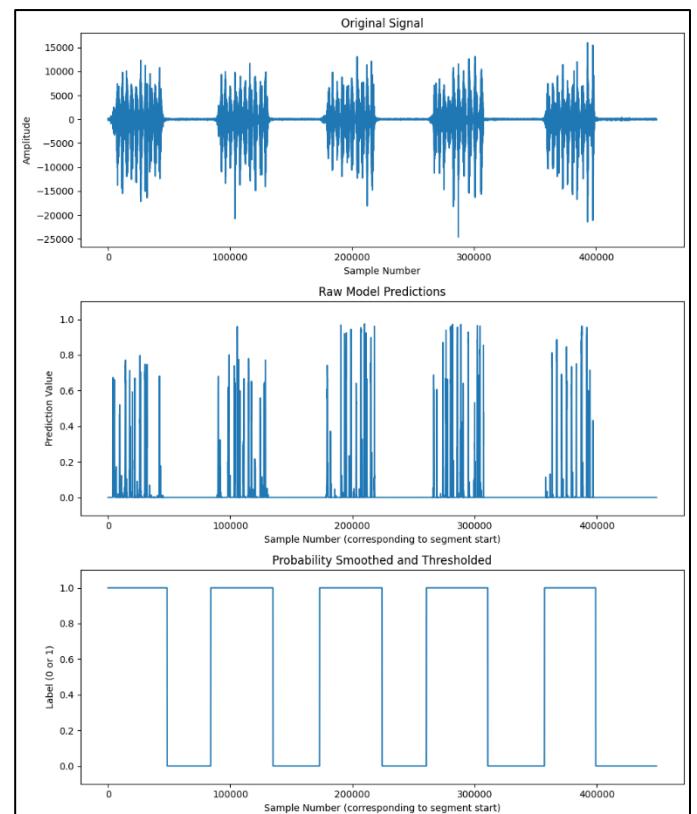


Figura 27 — EMG tríceps braquial procesado con CNN y postprocesamiento — ventana 250 muestras, umbral 0.01.

Los resultados obtenidos con el modelo CNN, tanto en señales del bíceps como en la proyección a músculos antagonistas como el tríceps, destacan su robustez y la efectividad del posprocesamiento con suavizado de probabilidad y umbralización en la reducción de falsos negativos, mejorando la detección de

estados musculares en entornos desbalanceados.

LSTM: se implementó para procesar secuencias derivadas de las características de Hjorth, tratadas como series temporales con una forma de entrada ajustada a (3, 1) por segmento para enfocarse en patrones secuenciales locales. A continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Keras (Tabla 23), la matriz de confusión (Figura 28) y el análisis de costo computacional (Tabla 24).

Clase	Precision	Recall	F1-Score	Soporte
0	0.9979	1.0000	0.9990	56 414
1	1.0000	0.9986	0.9993	85 187
Accuracy	—	—	0.9992	141 601
Macro Avg	0.9990	0.9993	0.9991	141 601
Weighted Avg	0.9992	0.9992	0.9992	141 601

Tabla 23. Resultados del modelo Long Short-Term Memory (LSTM).

Los resultados preliminares muestran un rendimiento excepcionalmente alto, superior a modelos anteriores en términos de precisión global y equilibrio entre clases.

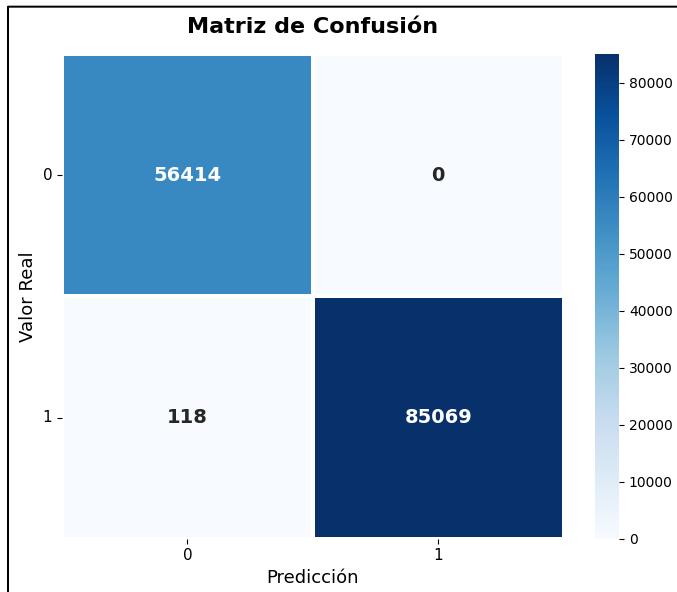


Fig. 28. Matriz de confusión Long Short-Term Memory (LSTM).

Adicionalmente, se estimó el costo computacional del modelo para valorar su aplicabilidad en entornos de tiempo real, considerando 100 iteraciones de inferencia.

Métrica	Valor
Inference time (per sample)	0.101683 ± 0.022840
RAM usage	1100.480469 MB
Model disk space	0.250885 MB

Tabla 24. Análisis de rendimiento Long Short-Term Memory (LSTM), basado en 100 iteraciones.

Pruebas

El modelo se evaluó utilizando señales EMG crudas, aplicando la misma cadena de preprocesamiento y extracción de características descrita en fases previas (procesado y optimización de la base de datos). Se observa un patrón similar a modelos anteriores, con una alta exactitud general, pero con una tendencia a niveles de probabilidad intermedios durante períodos de actividad muscular, indicando posible confusión en transiciones, donde el modelo tiende a subestimar muestras de contracción como reposo (véase Figura 29).

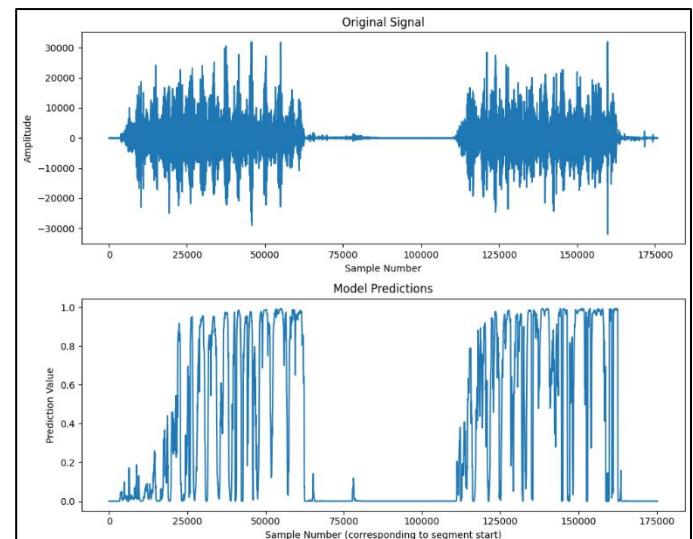


Fig. 29. Señal de entrada versus predicciones crudas del modelo LSTM con tasa de confusión elevada en transiciones.

El suavizado de probabilidad aplicado sobre las probabilidades de predicción reduce

significativamente los casos de confusión aislados, mejorando la continuidad temporal de las detecciones (Figura 30). Este método considera la influencia de predicciones vecinas mediante una media móvil con ventana de 150 muestras (~ 75 ms) y umbralización en 0.1, eliminando artefactos del vector de predicciones, alineándose con estrategias exitosas en modelos previos.

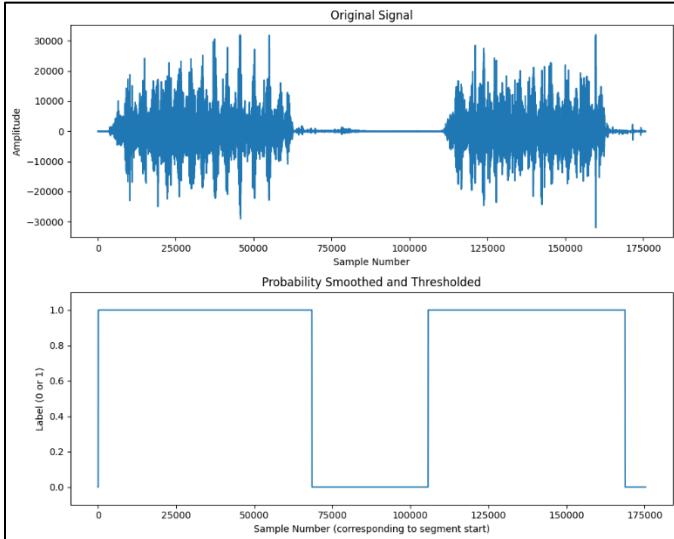


Fig. 30. Señal de entrada versus vector de predicciones LSTM después del postprocesamiento.

Dado que los resultados preliminares de modelos clásicos y CNN con la misma metodología sugieren una generalización efectiva al tríceps braquial, con una tasa de FN similar a las señales del bíceps durante contracciones, se espera un comportamiento análogo para el LSTM (ejemplo en la Figura 31), dada su capacidad para aprender dependencias temporales robustas a partir de las características de Hjorth.

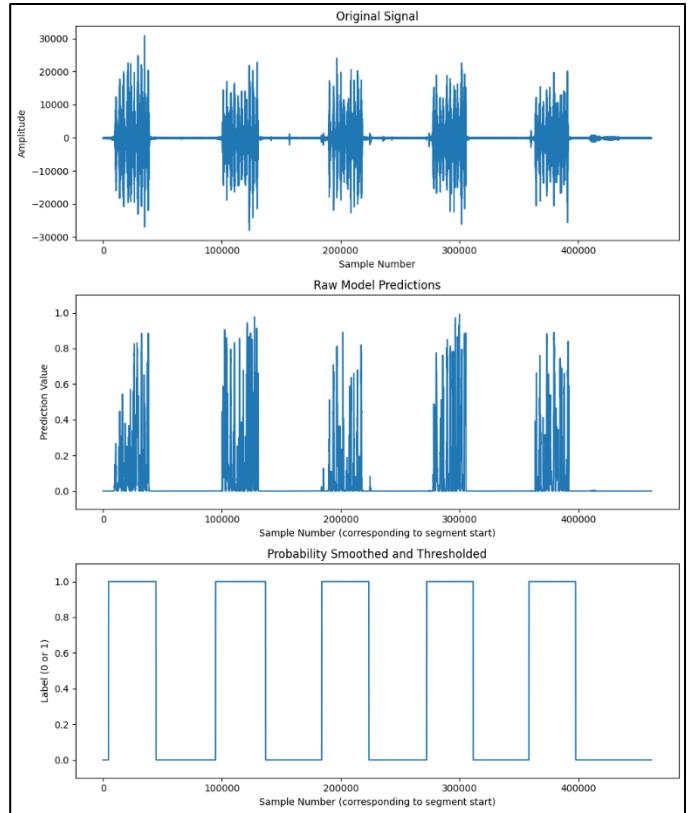


Fig. 31. EMG tríceps braquial procesado con LSTM y postprocesamiento, mostrando robustez en señales de extensión muscular (ventana 250 muestras, umbral 0.01).

Los resultados obtenidos con el modelo LSTM, tanto en señales del bíceps como en la proyección a músculos antagonistas como el tríceps, destacan su robustez y la efectividad del posprocesamiento con suavizado de probabilidad y umbralización en la reducción de falsos negativos, mejorando la detección de estados musculares en entornos desbalanceados.

CNN-LSTM: se implementó un modelo CNN-LSTM para procesar secuencias derivadas de las características de Hjorth, tratadas como series temporales con una forma de entrada ajustada a (3, 1) por segmento, aplicando convoluciones 1D para resaltar patrones locales antes de la capa recurrente. A continuación, se presentan los resultados obtenidos mediante el reporte de clasificación de Keras (Tabla 25), la matriz de confusión (Figura 33) y el análisis de costo computacional (Tabla 26).

Clase	Precisión	Recall	F1-score	Soporte
0	0.9980	0.9999	0.9989	56,414
1	0.9999	0.9987	0.9993	85,187
Exactitud			0.9992	141,601
Promedio macro	0.9990	0.9993	0.9991	141,601
Promedio ponderado	falta valor	falta valor	falta valor	141,601

Tabla 25. Resultados del modelo Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM).

Los resultados preliminares muestran un rendimiento excepcionalmente alto, superior a modelos anteriores en términos de precisión global y equilibrio entre clases.

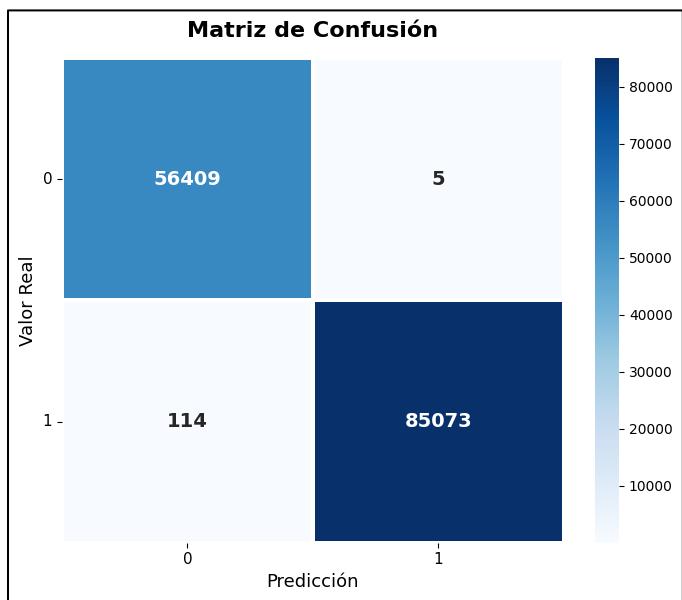


Fig. 33. Matriz de CNN-LSTM Memory (LSTM).

Adicionalmente, se estimó el costo computacional del modelo para valorar su aplicabilidad en entornos de tiempo real, considerando 100 iteraciones de inferencia.

Métrica	Valor
Tiempo de inferencia (promedio)	0.447545 ± 0.193231 S
Uso de RAM	~1.3 GB
Espacio en disco del modelo	~0.36 MB

Tabla 25. Análisis de rendimiento CNN-LSTM, basado en 100 iteraciones.

Pruebas

El modelo se evaluó utilizando señales EMG crudas, aplicando la misma cadena de preprocesamiento y extracción de características descrita en fases previas, se observa un patrón similar a modelos anteriores, con una alta exactitud general, pero con una tendencia a niveles de probabilidad intermedios durante períodos de actividad muscular, indicando posible confusión en transiciones, donde el modelo tiende a subestimar muestras de contracción como reposo (véase Figura 34).

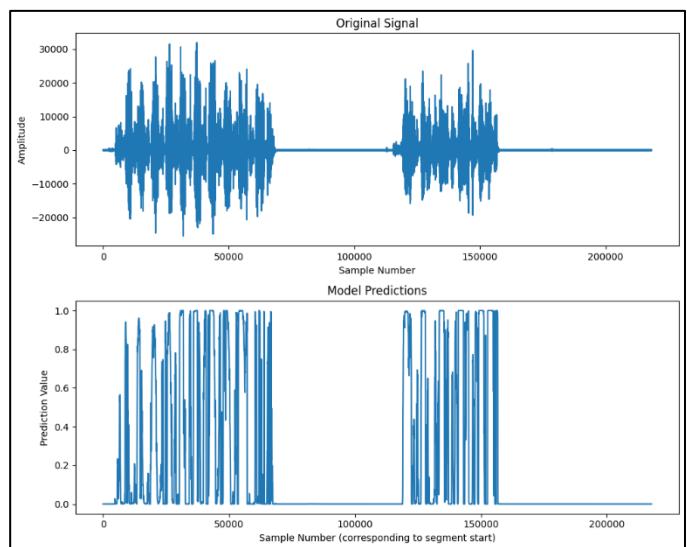


Fig. 34. Señal de entrada versus predicciones crudas del modelo CNN-LSTM con tasa de confusión elevada en transiciones.

El suavizado de probabilidad aplicado sobre las probabilidades de predicción reduce significativamente los casos de confusión aislados, mejorando la continuidad temporal de las detecciones (Figura 35). Este método considera la influencia de predicciones vecinas mediante una media móvil con ventana de 150 muestras (~75 ms) y umbralización en 0.01, eliminando artefactos del vector de predicciones, alineándose con estrategias exitosas en modelos previos.

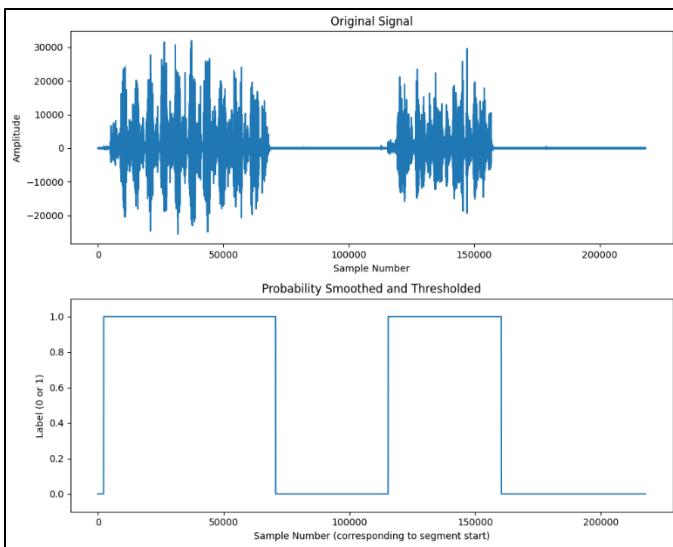


Fig. 35. Señal de entrada versus vector de predicciones CNN-LSTM después del postprocesamiento.

Dado que los resultados preliminares de modelos previos sugieren una generalización efectiva al tríceps braquial, se espera un comportamiento análogo para el CNN-LSTM (ejemplo en la Figura 36), dada su capacidad para aprender dependencias temporales robustas a partir de las características de Hjorth.

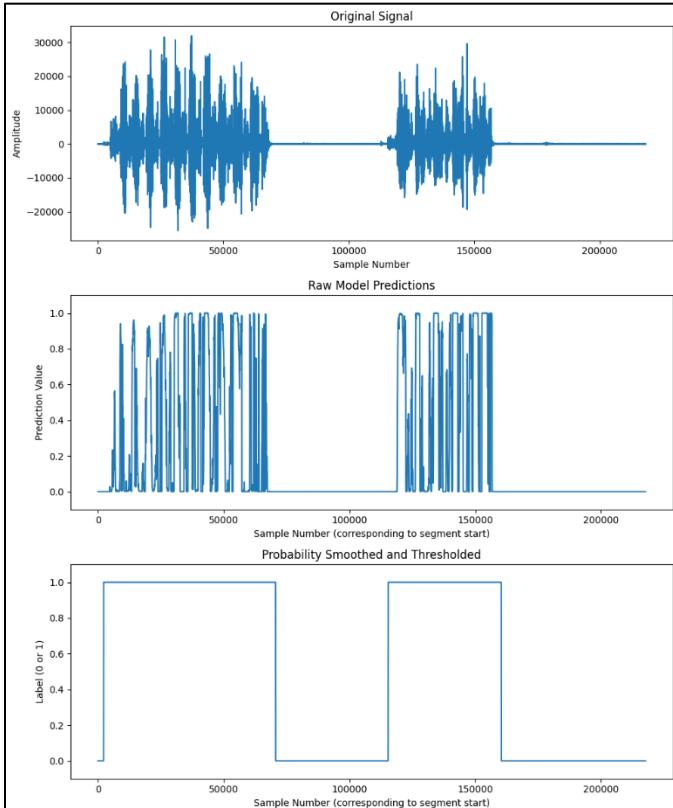


Fig. 36. EMG tríceps braquial procesado con CNN-LSTM y postprocesamiento, mostrando robustez en señales de extensión muscular (ventana 250 muestras, umbral 0.01).

Los resultados obtenidos con el modelo CNN-LSTM, tanto en señales del bíceps como en la proyección a músculos antagonistas como el tríceps, destacan su robustez y la efectividad del posprocesamiento con suavizado de probabilidad y umbralización en la reducción de falsos negativos, mejorando la detección de estados musculares en entornos desbalanceados.

Resumen de las pruebas DL

Esta sección presenta un pequeño resumen de análisis comparativo de los modelos de aprendizaje profundo, aquí se detallan los puntos clave:

- Generalización: Todos los modelos generalizan bien a tríceps, con altas tasas de falsos negativos principalmente observadas durante las pruebas de desempeño, estas fueron reducidas mediante técnicas de postprocesamiento que consistían en filtros de media móvil y umbralización.
- Mejoras: se observa una reducción importante en los tiempos de inferencia (~XX s), posiblemente debido que estos modelos consumen una cantidad más grande de recurso computacionales.
- Postprocesamiento: El suavizado con media móvil y umbralización mejora la detección, especialmente en datos desbalanceados, sin embargo.

Aunque los resultados son similares a modelos más sencillos (ML), su implementación para aplicaciones clínicas y deportivas, especialmente en sistemas de baja potencia no parece prometedora, no obstante, se pueden optimizar el desbalanceo de clases, así como los hiperparámetros mejorar el desempeño más el desempeño, veamos la tabla 25 y 26 donde se presentan un resumen de las métricas de cada modelo de aprendizaje profundo.

Modelo	Precisión	Recall	F1-Score	Exactitud	Tiempo de inferencia
CNN	0.9997	0.9997	0.9997	0.9997	0.15
LSTM	0.9992	0.9992	0.9992	0.9992	0.12
CNN-LSTM	0.9990*	0.9993*	0.9991*	0.9992	0.18

Tabla 25. Resumen de métricas ML (durante el entrenamiento).

Modelo	Uso de RAM	Espacio en disco del modelo	Viabilidad en tiempo real
CNN	1661.48 MB	0.0607 MB	Alta
LSTM	1100.48 MB	0.2509 MB	Media-Alta
CNN-LSTM	≈ 1.3 GB	≈ 0.36 MB	Baja

Tabla 26. Resumen de costo computacional modelos DL (durante el entrenamiento).

Una vez evaluados los modelos de aprendizaje profundo, se demostró que todos presentan altas precisiones (>99%).

Resumen general

En esta sección se presenta un resumen general del desempeño comparativo de todos los modelos evaluados (ML y DL) para la clasificación de estados musculares.

Todos los modelos fueron entrenados y validados utilizando el conjunto de datos optimizado emg_features_optimized.csv, con una partición del 80% para entrenamiento y 20% para validación, asegurando reproducibilidad y evaluación en métricas estándar de clasificación binaria, también se midió el tiempo de inferencia promedio por muestra para evaluar la viabilidad en aplicaciones de tiempo real, por ejemplo, monitoreo clínico o rehabilitación muscular.

Los resultados globales destacan un rendimiento elevado en todos los modelos (figura 37), con precisiones superiores al 96%, evidenciando la efectividad de las características

de Hjorth en la discriminación de patrones electrofisiológicos.

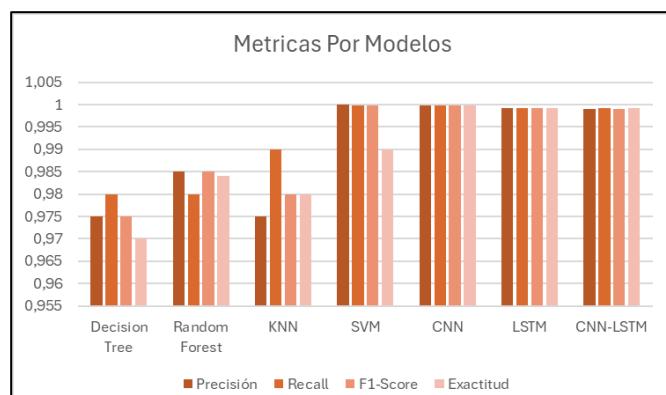


Fig. 37: Exactitud (accuracy), precisión (precision), exhaustividad (recall) y puntaje F1 por modelo.

El modelo Random Forest resultó ser el más equilibrado, alcanzando una exactitud del 99% para la clase de actividad y 98% para reposo, con un tiempo de inferencia mínimo de ~0.04 segundos, (figura 38) superando otros métodos tradicionales basados en umbrales y demostrando robustez frente a artefactos residuales mediante el posprocesamiento propuesto.

En contraste, los modelos profundos como LSTM y CNN-LSTM exhibieron precisiones similares o ligeramente superiores (figura 38), pero con tiempos de inferencia más elevados (~0.1-0.2 segundos), atribuidos a su mayor complejidad computacional, esto los hace ideales para escenarios con dependencias temporales complejas, pero menos eficientes en sistemas embebidos de baja potencia.

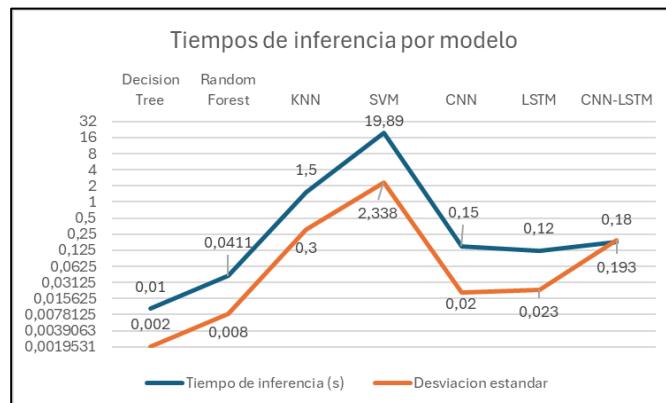


Fig. 38: Tiempo de inferencia por modelo.

Los enfoques clásicos como DT y KNN ofrecieron un compromiso favorable entre precisión (97-98%) y eficiencia, mientras que SVM presentó el mayor costo computacional (figura 39 y 40), posiblemente debido a sobreajuste en el espacio de características reducido.

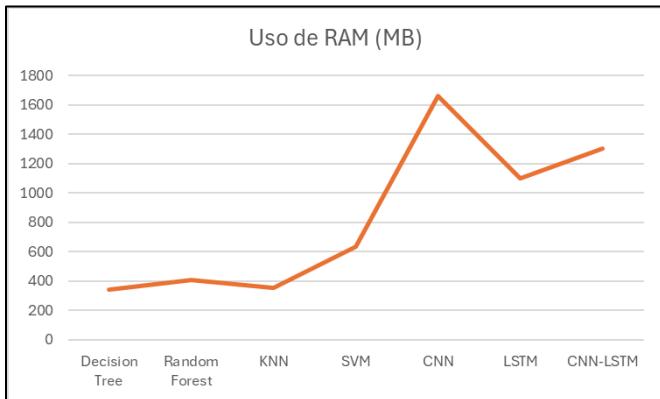


Fig. 39: Uso de RAM por modelo.

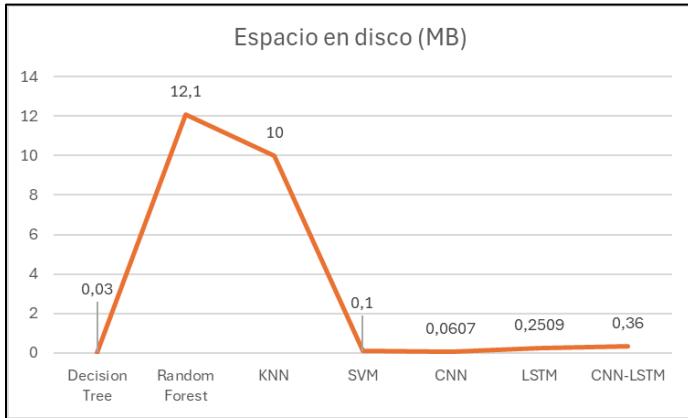


Fig. 40: Espacio en disco por modelo.

La generalización a señales de músculos antagonistas, fue consistente en todos los modelos, con reducciones significativas en tasas de falsos negativos (FN) tras el posprocesamiento, confirmando la versatilidad del enfoque para aplicaciones multimúsculo mediante ajuste fino de los parámetros width y threshold, a continuación se presenta un resumen general de todas las métricas por modelo en las figuras 27 y 28.

Modelo	Precisión	Recall	F1-Score	Exactitud
Decision Tree	0,975	0,98	0,975	0,97
Random Forest	0,985	0,98	0,985	0,984
KNN	0,975	0,99	0,98	0,98

Modelo	Precisión	Recall	F1-Score	Exactitud
SVM	0,9999	0,9997	0,9998	0,99
CNN	0,9997	0,9997	0,9997	0,9997
LSTM	0,9992	0,9992	0,9992	0,9992
CNN-LSTM	0,999	0,9993	0,999	0,9992

Tabla 27: Resumen comparativo de métricas de todos los modelos.

Modelo	Tiempo de inferencia (s)	Desviación (S)	Uso de RAM (MB)	Espacio en disco (MB)
Decision Tree	0,01	0,002	342	0,03
Random Forest	0,0411	0,008	405	12,1
KNN	1,5	0,3	356	10
SVM	19,89	2,338	637	0,1
CNN	0,15	0,02	1661,48	0,0607
LSTM	0,12	0,023	1100,48	0,2509
CNN-LSTM	0,18	0,193	1300	0,36

Tabla 28. costos computacionales de todos los modelos.

proyección a futuro

Posibles mejoras en la base de datos

- Realizar modificaciones a la base de datos para adaptar los datos a las arquitecturas DL (Deep Learning), lo que consiste en una representación simple para CNN y una representación secuencial LSTM y CNN-LSTM, además de algunas optimizaciones en el etiquetado automático mediante las técnicas vistas durante el postprocesado de las pruebas anteriores.
- Realizar ajustes a la base de datos para optimizarla y adaptarla a los requerimientos de los modelos de aprendizaje profundo (DL), partiendo de los registros preprocesados en las fases iniciales de desarrollo para recalcular las características de Hjorth y el uso de etiquetas automáticas derivadas de K-means, podemos ver un ejemplo en la figura 23.

3. Para el caso de LSTM y CNN-LSTM antes de unir todos los registros se crearon arreglos de 6 ventanas consecutivas que posteriormente servirán de entradas para el modelo.
4. una revisión y exportación individual por registro en formato CSV para facilitar el manejo modular y escalable, posteriormente se combinaron todos los registros CSV en uno solo para aplicar la optimización de tipo de variable.

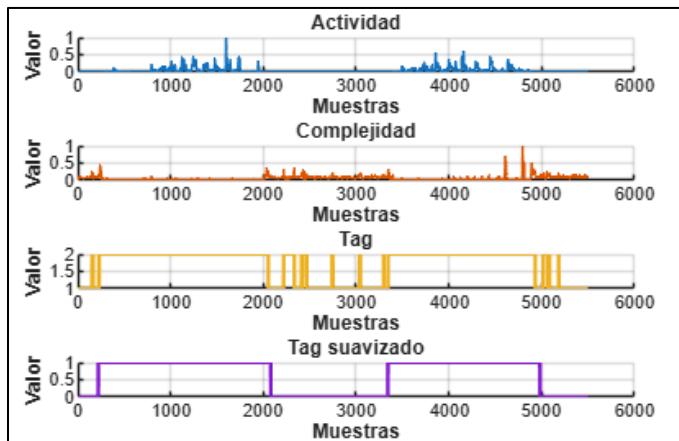


Fig. 23. Parámetros de Hjorth (actividad, complejidad) y etiquetado automático antes y después de procesar el vector actividad.

Como podemos en el ejemplo observar esto mitiga errores aislados en transiciones musculares o causados por ruido o desbalance, este post-clustering temporal mejora la coherencia del etiquetado con ventanas de 50-500 muestras, lo que reduce dramáticamente los falsos negativos en el conjunto de entrenamiento.

Ventajas y limitaciones

El análisis comparativo de estos modelos permitió identificar las fortalezas y limitaciones más relevantes de cada enfoque, tanto en términos de desempeño como de costo computacional.

Como pudimos observar en la Tabla 27 los modelos basados en aprendizaje automático ofrecen tiempos de respuesta más reducidos y menor demanda de recursos, mientras que los basados en aprendizaje profundo ofrecen

tiempos de respuesta similares y mayor precisión a costa de un mayor requerimiento de recursos computacionales, ahora veamos un desglose de las ventajas y limitaciones.

Ventajas

- Decision Tree y Random Forest presentan una alta interpretabilidad, bajo consumo de memoria y excelente viabilidad en tiempo real, lo que los hace adecuados para implementaciones embebidas o sistemas de monitoreo continuo.
- SVM logra una precisión casi perfecta ($\approx 0,9999$), destacándose en la clasificación de patrones bien delimitados, aunque su costo computacional es considerable.
- CNN, LSTM y CNN-LSTM muestran una capacidad superior de generalización, alcanzando métricas cercanas y una mayor robustez ante señales con altas relaciones de señal-ruido.

Limitaciones

- KNN evidencia una baja viabilidad en tiempo real, debido a su dependencia directa del conjunto de entrenamiento durante la inferencia.
- SVM, a pesar de su alta precisión, requiere elevados tiempos de inferencia y un consumo significativo de memoria, lo que limita su aplicación en entornos con recursos restringidos.
- Los modelos DL (CNN, LSTM y CNN-LSTM), aunque precisos, presentan mayor complejidad computacional, tiempos de inferencia más altos y un proceso de entrenamiento prolongado.
- CNN-LSTM, en particular, demanda una mayor capacidad de procesamiento debido a la combinación de capas convolucionales y recurrentes.

En resumen, los modelos de aprendizaje automático se perfilan como opciones eficientes para sistemas en tiempo real, mientras que los modelos profundos resultan más adecuados para escenarios de alta precisión y análisis detallado de señales EMG.

Comparación con literatura

En general, los modelos de DL superan a los de ML clásicos en precisión de clasificación EMG. Por ejemplo, Moyassar y Abdullah [29] obtuvieron precisión de 99.81% con Random Forest (ML) y 99.61% con CNN (DL), destacando que sus métodos logran alta precisión con un bajo tiempo de computación.

Sevilla-Camacho et al. [30] comparan LSTM frente a SVM y reportan 100% de precisión con un LSTM (fase 1) versus 93% con SVM clásico.

Zaim et al. [31] encontraron que KNN y Random Forest alcanzan ~99% de exactitud en tareas mioeléctricas, mientras que SVM se queda en ~86.9%

En contraste, Zha et al. [32], en un entorno intraoperatorio con ruido, sólo obtuvieron ~89.5% con un modelo híbrido CNN-LSTM, sugiriendo que la calidad del dato influye fuertemente en el desempeño.

Aunque no en todos estos trabajos se presentaron de manera explícita, los F1-scores siguen la misma tendencia, reflejando balances de precisión/recall similares.

En general, los modelos de DL suelen requerir más recursos, pero se pueden optimizar. Sevilla-Camacho et al. [30] midieron tiempos de inferencia: un LSTM simple clasifica una muestra en ≈ 0.12 ms, frente a 2.7 ms con un SVM convencional

Choi [33] diseñó una CNN ligera para EMG sobre FPGA con ~8122 parámetros entrenables, logrando ≈ 0.12 ms por muestra en hardware embebido, este diseño alcanzó ~96% de

precisión a pesar de ser de arquitectura ligera, evidenciando que es factible implementar redes DL eficientes en tiempo real.

En conjunto, la literatura reciente indica que las redes profundas (CNN, LSTM, CNN-LSTM) proporcionan una alta precisión a expensas de mayor complejidad, aunque arquitecturas clásicas y ligeras pueden optimizarse para acercarse a la eficiencia de los clasificadores DL que en general muestran precisiones también, pero con tiempos de inferencia típicamente mayores.

Posibles aplicaciones

La clasificación precisa de señales electromiográficas (EMG) mediante modelos de aprendizaje automático y profundo abre un amplio espectro de aplicaciones prácticas, especialmente en el ámbito biomédico, a continuación, se plantean algunas áreas donde estos pueden ser implementados:

- Desarrollo de prótesis controladas por señales EMG, este tipo de clasificación permite un control más natural y fluido de extremidades artificiales, aumentando la autonomía de personas con amputaciones.
- Detección en tiempo real de la actividad muscular puede emplearse para monitorizar el progreso en terapias de rehabilitación motora, indicar dinámicamente el nivel de asistencia requerido y proporcionar retroalimentación al especialista médico concerniente.
- La integración de señales EMG como canal de entrada en interfaces multimodales permite desarrollar soluciones de asistencia para personas con movilidad reducida.
- En el campo del entretenimiento interactivo, las señales musculares pueden utilizarse para generar comandos naturales en videojuegos o experiencias inmersivas,

favoreciendo la inclusión de usuarios con discapacidades motrices.

Conclusiones

Se observó que los modelos CNN-LSTM y LSTM son particularmente efectivos en tareas que requieren la modelación de características espaciales y temporales, lo cual se refleja en su capacidad para generalizar patrones complejos presentes en la señal EMG. Sin embargo, estas arquitecturas también presentan un mayor costo computacional, tanto en tiempo de inferencia como en uso de memoria, lo cual limita su aplicabilidad en sistemas embebidos o dispositivos portátiles sin técnicas de optimización adicionales.

La comparación con literatura reciente respalda esto, confirmando que las redes neuronales profundas, cuando son adecuadamente diseñadas o desplegadas en hardware optimizado (como FPGAs), pueden alcanzar precisiones similares o superiores a los métodos tradicionales con un costo computacional controlado.

Adicionalmente, algunos de estos estudios demuestran que es posible implementar modelos DL ligeros con latencias competitivas, lo cual sugiere la posibilidad de su integración en dispositivos embebidos de baja potencia.

No obstante, la elección del modelo adecuado depende del compromiso entre precisión requeridos, los recursos disponibles y las condiciones operativas.

Los modelos DL se perfilan como la mejor alternativa en aplicaciones donde se prioriza la precisión y robustez, mientras que los modelos ML siguen siendo válidos en escenarios donde la eficiencia computacional y la interpretabilidad son fundamentales.

REFERENCIAS

- [1] Z. Hussain, A. A. Gopalai, S. A. Ahmad, M. S. B. Salim, D. Gouwanda, and P.-L. Teh, “Investigating age-related muscle force adaptations in males and females during sit-to-walk transition motion using EMG-informed modeling,” *Results in Engineering*, vol. 26, p. 104660, Jun. 2025, doi: 10.1016/j.rineng.2025.104660.
- [2] MathWorks, “audioread - Read audio file - MATLAB,” 2025. Accessed: Sep. 30, 2025. [Online]. Available: <https://la.mathworks.com/help/matlab/ref/audioread.html>
- [3] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan, and T. Chang, “Surface Electromyography Signal Processing and Classification Techniques,” *Sensors*, vol. 13, no. 9, pp. 12431–12466, Sep. 2013, doi: 10.3390/s130912431.
- [4] Y.-D. Wu, S.-J. Ruan, and Y.-H. Lee, “An Ultra-Low Power Surface EMG Sensor for Wearable Biometric and Medical Applications,” *Biosensors (Basel)*, vol. 11, no. 11, p. 411, Oct. 2021, doi: 10.3390/bios11110411.
- [5] M. Boyer, L. Bouyer, J.-S. Roy, and A. Campeau-Lecours, “Reducing Noise, Artifacts and Interference in Single-Channel EMG Signals: A Review,” *Sensors*, vol. 23, no. 6, p. 2927, Mar. 2023, doi: 10.3390/s23062927.
- [6] X. Li, W. Z. Rymer, G. Li, and P. Zhou, “The effects of notch filtering on electrically evoked myoelectric signals and associated motor unit index estimates,” *J Neuroeng Rehabil*, vol. 8, no. 1, p. 64, Dec. 2011, doi: 10.1186/1743-0003-8-64.
- [7] L. Shaw and S. Bhaga, “Online EMG Signal Analysis for Diagnosis of Neuromuscular Diseases by Using PCA and PNN,” *International Journal of Engineering Science and Technology (IJEST)*, vol. 4, no. 10, pp. 4453–4459, 2012, [Online]. Available: https://www.researchgate.net/publication/232905752_Online_EMG_Signal_Analysis_for_diagnosis_of_Neuromuscular_diseases_by_using_PCA_and_PNN
- [8] N. B. M. Voet, C. G. J. Saris, D. H. J. Thijssen, V. Bastiaans, D. E. Sluijs, and M. M. H. P. Janssen, “Surface Electromyography Thresholds as a Measure for Performance Fatigability During Incremental Cycling in Patients With Neuromuscular Disorders,” *Front Physiol*, vol. 13, Mar. 2022, doi: 10.3389/fphys.2022.821584.
- [9] A. Sarmento *et al.*, “Changes in electromyographic activity, mechanical power, and relaxation rates following inspiratory ribcage muscle fatigue,” *Sci Rep*, vol. 11, no. 1, p. 12475, Jun. 2021, doi: 10.1038/s41598-021-92060-y.

- [10] R. So, K.-M. Chan, and O. Siu, “EMG Power Frequency Spectrum Shifts during Repeated Isokinetic Knee and Arm Movements,” *Res Q Exerc Sport*, vol. 73, no. 1, pp. 98–106, Mar. 2002, doi: 10.1080/02701367.2002.10608996.
- [11] A. Sharma, “A Comparative Study of Different EMG Features for Acoustics-to-EMG Mapping,” in *Proceedings of Interspeech 2021*, Brno, Czech Republic, 2021. [Online]. Available: https://www.isca-archive.org/interspeech_2021/sharma21_interspeech.pdf
- [12] A. Rizal, S. Hadiyoso, and A. Z. Ramdani, “FPGA-Based Implementation for Real-Time Epileptic EEG Classification Using Hjorth Descriptor and KNN,” *Electronics (Basel)*, vol. 11, no. 19, p. 3026, Sep. 2022, doi: 10.3390/electronics11193026.
- [13] W. H. Alawee, A. Basem, and L. A. Al-Haddad, “Advancing biomedical engineering: Leveraging Hjorth features for electroencephalography signal analysis,” *J Electr Bioimpedance*, vol. 14, no. 1, pp. 66–72, Jan. 2023, doi: 10.2478/joeb-2023-0009.
- [14] D. SASIDHARAN, G. VENUGOPAL, and R. SWAMINATHAN, “COMPLEXITY ANALYSIS OF SURFACE ELECTROMYOGRAPHY SIGNALS UNDER FATIGUE USING HJORTH PARAMETERS AND BUBBLE ENTROPY,” *J Mech Med Biol*, vol. 23, no. 06, Aug. 2023, doi: 10.1142/S0219519423400511.
- [15] H. da Rocha, J. Pereira, R. Abrishambaf, and A. Espirito Santo, “An Interoperable Digital Twin with the IEEE 1451 Standards,” *Sensors*, vol. 22, no. 19, p. 7590, Oct. 2022, doi: 10.3390/s22197590.
- [16] J. Y. Lee, J. Hur, J. Choi, R.-H. Park, and J. Kim, “Multi-scale foreground-background separation for light field depth estimation with deep convolutional networks,” *Pattern Recognit Lett*, vol. 171, pp. 138–147, Jul. 2023, doi: 10.1016/j.patrec.2023.05.014.
- [17] T. Tanaka, I. Nambu, Y. Maruyama, and Y. Wada, “Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography,” May 2022, doi: 10.3390/s22135005.
- [18] P. Y. Jayaweera, “Design and Implementation of Electromyography (EMG) based Real-Time Pattern Recognition model for Prosthetic hand Control,” Sep. 28, 2021. doi: 10.31219/osf.io/rd2cf.
- [19] L. McManus, G. De Vito, and M. M. Lowery, “Analysis and Biophysics of Surface EMG for Physiotherapists and Kinesiologists: Toward a Common Language With Rehabilitation Engineers,” *Front Neurol*, vol. 11, Oct. 2020, doi: 10.3389/fneur.2020.576729.
- [20] M. J. Islam *et al.*, “Myoelectric Pattern Recognition Performance Enhancement Using Nonlinear Features.,” *Comput Intell Neurosci*, vol. 2022, p. 6414664, 2022, doi: 10.1155/2022/6414664.
- [21] A. Sultana, F. Ahmed, and Md. S. Alam, “A systematic review on surface electromyography-based classification system for identifying hand and finger movements,” *Healthcare Analytics*, vol. 3, p. 100126, Nov. 2023, doi: 10.1016/j.health.2022.100126.
- [22] Y.-Y. Song and Y. Lu, “Decision tree methods: applications for classification and prediction.,” *Shanghai Arch Psychiatry*, vol. 27, no. 2, pp. 130–5, Apr. 2015, doi: 10.11919/j.issn.1002-0829.215044.

- [23] L. Breiman, “Random Forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [24] B. Sun and H. Chen, “A Survey of k Nearest Neighbor Algorithms for Solving the Class Imbalanced Problem,” *Wirel Commun Mob Comput*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5520990.
- [25] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [26] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning,” *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, Sep. 2020, doi: 10.1007/s11831-019-09344-w.
- [27] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [28] L. Liu, J. Feng, J. Li, W. Chen, Z. Mao, and X. Tan, “Multi-layer CNN-LSTM network with self-attention mechanism for robust estimation of nonlinear uncertain systems.,” *Front Neurosci*, vol. 18, p. 1379495, 2024, doi: 10.3389/fnins.2024.1379495.
- [29] R. A. Moyassar and M. A. M. Abdullah, “Decoding Muscular Signals: Machine Learning Approaches to EMG Classification,” *Computing Open*, vol. 03, Jan. 2025, doi: 10.1142/S2972370125500047.
- [30] M. Aviles, J. M. Alvarez-Alvarado, J.-B. Robles-Ocampo, P. Y. Sevilla-Camacho, and J. Rodríguez-Reséndiz, “Optimizing RNNs for EMG Signal Classification: A Novel Strategy Using Grey Wolf Optimization,” *Bioengineering*, vol. 11, no. 1, p. 77, Jan. 2024, doi: 10.3390/bioengineering11010077.
- [31] T. Zaim, S. Abdel-Hadi, R. Mahmoud, A. Khandakar, S. M. Rakhtala, and M. E. H. Chowdhury, “Machine Learning- and Deep Learning-Based Myoelectric Control System for Upper Limb Rehabilitation Utilizing EEG and EMG Signals: A Systematic Review,” *Bioengineering*, vol. 12, no. 2, p. 144, Feb. 2025, doi: 10.3390/bioengineering12020144.
- [32] X. Zha *et al.*, “A Deep Learning Model for Automated Classification of Intraoperative Continuous EMG,” *IEEE Trans Med Robot Bionics*, vol. 3, no. 1, pp. 44–52, Feb. 2021, doi: 10.1109/TMRB.2020.3048255.
- [33] H.-S. Choi, “Electromyogram (EMG) Signal Classification Based on Light-Weight Neural Network with FPGAs for Wearable Application,” *Electronics (Basel)*, vol. 12, no. 6, p. 1398, Mar. 2023, doi: 10.3390/electronics12061398.