**Logotipo

Descripción generada automáticamente**

**Reconocimiento de posturas de mano del dataset Ninapro usando aprendizaje automático**

Jairo Alberto Agudelo Medina

Henry Alberto Arcila Ramírez

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor  
Daniel Escobar Saltaren, Magíster en Ingeniería

Universidad de Antioquia  
Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2025

|  |  |
| --- | --- |
| **Cita** | (Agudelo Medina & Arcila Ramírez, 2025) |
| **Referencia**  **Estilo IEEE (2022)** | Agudelo Medina, J. A., & Arcila Ramírez, H. A. (2025). *Reconocimiento de posturas de mano del dataset Ninapro usando aprendizaje automático* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia. |

**** 

Grupo de Investigación Intelligent Information Systems Lab In2Lab

Especialización en Analítica y Ciencia de Datos, CohorteVIII.

Centro de Investigación Ambientales y de Ingeniería (CIA).

|  |  |
| --- | --- |
|  | Diagrama  Descripción generada automáticamente con confianza media |

Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

**Rector:** John Jairo Arboleda Céspedes.

**Decano:** Julio Cesar Saldarriaga Molina

**Jefe departamento:** Danny Alexandro Múnera Ramírez

**Coordinadora del Programa:** Maria Bernarda Salazar Sánchez

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

**Dedicatoria**

A nuestras familias, que se aguantan los tiempos de abandono.

**Agradecimientos**

A la Universidad de Antioquia por darnos nuevamente la oportunidad de estudiar.

**Tabla de contenido**

[Resumen 9](#_Toc199527718)

[1. Introducción 10](#_Toc199527719)

[2. Materiales y Métodos 12](#_Toc199527720)

[2.1. Fuente de datos 12](#_Toc199527721)

[2.2. Procesamiento de las señales musculares 13](#_Toc199527722)

[2.3. Modelos de inteligencia artificial 13](#_Toc199527723)

[2.4. Métricas 14](#_Toc199527724)

[3. Extracción de características 15](#_Toc199527725)

[4. Resultados 17](#_Toc199527726)

[5. Conclusiones 17](#_Toc199527727)

[Referencias 17](#_Toc199527728)

**Lista de Figuras**

[Figura 1. Tipos de agarres explorados. 12](#_Toc199694184)

[Figura 2. Señales de interés para un sujeto cualquiera. Arriba) Señal sEMG para 10 canales simultáneos. Medio) Numero de postura para cada una de las repeticiones, mencionado por los autores como “restimulus”. Abajo) Numero de repetición para cada uno de las posturas, mencionado por los autores como “rerepetition”. 13](#_Toc199694185)

[Figura 3. Muestra señales sEMG sujeto 1 15](#_Toc199694186)

[Figura 4. Posturas seleccionadas 15](#_Toc199694187)

[Figura 5. Método de preprocesamiento y extracción de características. 16](#_Toc199694188)

[Figura 6. Datos crudos del sujeto 1 para las posturas de interés. 16](#_Toc199694189)

[Figura 7. Señales sEMG crudas y filtradas 17](#_Toc199694190)

[Figura 8. Arquitectura red CNN-LSTM propuesta por Lin Chen et Al [14] 21](#_Toc199694191)

[Figura 9. Matriz de correlación para RMS 26](#_Toc199694192)

[Figura 10. Matriz de correlación para WL 26](#_Toc199694193)

[Figura 11. Matriz de correlación para IAV 27](#_Toc199694194)

[Figura 12. Matrices de confusión RMS, WL e IAV para el modelo SVM 27](#_Toc199694195)

[Figura 13. Matrices de confusión RMS, WL e IAV para el modelo KNN 28](#_Toc199694196)

[Figura 14. Matrices de confusión RMS, WL e IAV para el modelo RF 28](#_Toc199694197)

[Figura 15. Curvas ROC de RMS, WL e IAV para el modelo SVM 28](#_Toc199694198)

[Figura 16. Curvas ROC de RMS, WL e IAV para el modelo KNN 29](#_Toc199694199)

[Figura 17.Curvas ROC de RMS, WL e IAV para el modelo RF 29](#_Toc199694200)

[Figura 18. Matrices de confusión RMS, WL e IAV para el modelo CNN-LTSM 30](#_Toc199694201)

[Figura 19.Curvas ROC de RMS, WL e IAV para el modelo CNN-LTSM 30](#_Toc199694202)

**Lista de Tablas**

[Tabla 1. Características dataset ejercicio 3 de los 27 sujetos. 14](#_Toc199694322)

[Tabla 2. Dataset de características de las señales sEMG para las posturas elegidas. 18](#_Toc199694323)

[Tabla 3. Resumen de las características de los modelos probados. 19](#_Toc199694324)

[Tabla 4. Parámetros para SVM 19](#_Toc199694325)

[Tabla 5. Parámetros para KNN 20](#_Toc199694326)

[Tabla 6. Parámetros para RF 20](#_Toc199694327)

[Tabla 7. Arquitectura de red para CNN-LTSM 21](#_Toc199694328)

[Tabla 8.Hiperparámetros a optimizar 24](#_Toc199694329)

[Tabla 9. Resultados de los modelos 24](#_Toc199694330)

**Siglas, acrónimos y abreviaturas**

**IHM** Interfaz humano máquina

**ML** Machine Learning

**DL** Deep Learning

**ANN** Artificial Neural Network

**CNN** Convolutional Neural Network

**SVM** Support Vector Machines

**LDA** Linear Discriminant Analysis

**RF** Random forest

**KNN** K-Nearest Neighbors

**PCA** Principal Component Analysis

**ICA** Independent Component Analysis

**RBF** Radial Basis Function

**sEMG** Surface Electromyography

**RNN** Recurrent neural networks

**RMS** Root Mean Square

**WL** Waveform Length

**IAV** Integrated Absolute Value

**LSTM** Long Short-Term Memory Network

# Resumen

La pérdida de una extremidad corporal superior afecta significativamente la calidad de vida de quien lo padece. Para abordar esto se ha impulsado el desarrollo de prótesis, tanto comerciales como de código abierto, habilitando el acceso a esta tecnología a los pacientes. Un aspecto clave es el desarrollo del sistema de control, que involucra diseño electrónico, adquisición de datos y pruebas clínicas.

Los investigadores pueden utilizar bases de datos públicas como Ninapro y CapgMyo, que contienen señales electromiográficas, para entrenar modelos de aprendizaje automático y aprendizaje profundo, esenciales para el desarrollo de los controles. En este campo, debido a las limitaciones de los microcontroladores utilizados, técnicas clásicas de aprendizaje de máquina como máquinas de vectores de soporte, bosque aleatorio o k-vecinos más cercanos siguen siendo relevantes por su simplicidad y bajo requerimiento de recursos.

En este artículo se compara la capacidad de modelos de aprendizaje de máquina y aprendizaje profundo para clasificar dos tipos de agarre (movimientos 1 y 23) y el estado de reposo (movimiento 0) de la base de datos Ninapro DB1, utilizando como datos de entrada a los modelos, características extraídas las señales de electromiografía como raíz cuadrática media, longitud de forma de onda y valor absoluto integrado.

***Palabras claves:*** *Señales electromiográficas, Machine learning, Deep learning, NinaPro, modelos de clasificación, Root Mean Square, Waveform Length, Integrated Absolute Value.*

# Abstract

La pérdida de una extremidad corporal superior afecta significativamente la calidad de vida de quien lo padece. Para abordar esto se ha impulsado el desarrollo de prótesis, tanto comerciales como de código abierto, habilitando el acceso a esta tecnología a los pacientes. Un aspecto clave es el desarrollo del sistema de control, que involucra diseño electrónico, adquisición de datos y pruebas clínicas.

Los investigadores pueden utilizar bases de datos públicas como Ninapro y CapgMyo, que contienen señales electromiográficas, para entrenar modelos de aprendizaje automático y aprendizaje profundo, esenciales para el desarrollo de los controles. En este campo, debido a las limitaciones de los microcontroladores utilizados, técnicas clásicas de aprendizaje de máquina como máquinas de vectores de soporte, bosque aleatorio o k-vecinos más cercanos siguen siendo relevantes por su simplicidad y bajo requerimiento de recursos.

En este artículo se compara la capacidad de modelos de aprendizaje de máquina y aprendizaje profundo para clasificar dos tipos de agarre (movimientos 1 y 23) y el estado de reposo (movimiento 0) de la base de datos Ninapro DB1, utilizando como datos de entrada a los modelos, características extraídas las señales de electromiografía como raíz cuadrática media, longitud de forma de onda y valor absoluto integrado.

***Keywords:*** *Señales electromiográficas, machine learning, NinaPro, modelos de clasificación, Root Mean Square.*

# Introducción

La pérdida de una extremidad superior es un problema que afecta la calidad de vida de cualquier persona, ya que tras una amputación su vida normal, independencia, funcionalidad, imagen de sí misma y su salud no solo física sino mental se ven comprometidas [1]. Con el fin de mejorar la calidad de vida perdida tras una amputación el campo Interfaz Hombre-Máquina (HMI, del inglés Human Machine Interface), ha impulsado la investigación y desarrollo de prótesis tanto comerciales [2], [3], [4] como de código abierto [5], [6], [7] lo que ha contribuido a la democratización del acceso a esta tecnología, especialmente en contextos académicos y de investigación aplicada. Uno de los elementos claves de una prótesis es la concepción y diseño del sistema de control pues involucra no solo tareas de diseño electrónico, sino también el desarrollo de protocolos de adquisición de datos y pruebas clínicas [8].

El desarrollo de pruebas clínicas y de ingeniería involucran procesos de adquisición de datos que son costosos debido a las limitaciones en recursos humanos y tecnológicos. Afortunadamente existen datasets públicos como Ninapro [9] o CapgMyo [10] que comparten registros de diferentes señales electromiográficas. Estas bases de datos son ampliamente usadas por   la comunidad científica como insumo para el desarrollo de modelos de aprendizaje automático, (ML, del inglés Machine Learning) involucrados en el diseño de los sistemas de control de prótesis [11] que apuntan a mejorar la experiencia de usuario y la adherencia de estos.

En aprendizaje automático la extracción de características es una etapa fundamental en la que se busca capturar la mayor cantidad posible de información, con características de temporalidad y frecuencia, que luego será llevada a los modelos de clasificación y/o de clustering para entrenamiento y validación. Utilizar un conjunto de características cuidadosamente seleccionado puede proporcionar una tasa de clasificación más alta que la señal cruda [12]. Múltiples técnicas temporales relacionadas en [2b-c] y frecuenciales relacionadas en [2c] son utilizadas para extraer las características.

De igual forma la reducción de la dimensionalidad es a menudo necesaria especialmente en dominios tiempo-frecuencia de alta dimensionalidad. Se usa para reducir la carga del clasificador y/o el clustering y el tiempo computacional al reducir los datos a un espacio vectorial de baja dimensión mientras conservan la máxima información de la señal. La técnica más ampliamente usada es Análisis de Componentes Principales (PCA, del inglés - Principal Component Analysis).

Actualmente los desarrolladores de controles automáticos tienen algunas dificultades con el uso de sEMG, una de ella es la base de datos utilizada en el entrenamiento y validación, que a su vez depende de la cantidad de sensores usados, la posición de estos y el estado de salud y ánimo de los sujetos muestreados, etc.; otra es la escogencia de la técnica de ML o DL (del inglés Deep Learning), aprendizaje profundo, adecuada de entre las múltiples opciones que existen; y otra es la capacidad de los microcontroladores involucrados, generalmente con reducida memoria. Esto supone un gran reto para ellos.

El objetivo planteado en este trabajo es, usando las señales de EMG disponibles en la base de datos DB1 de Ninapro, comparar la capacidad de clasificación de diferentes de modelos de clasificación de DL y ML de agarres de mano a través del uso de las características individuales extraídas. Se trabajará con 2 tipos de agarre de la base de datos: 1, 23, y la posición de reposo, 0. Se quiere validar como varía la efectividad de los modelos de clasificación cuando se usa una cantidad específica de agarres, y no todos los disponible en la base de datos, y una característica específica extraída de las señales. Esto en linea con el uso real de las prótesis comerciales, que por lo general utilizan uno o dos agarres (en configuraciones de apertura y cierre) y a lo sumo la rotación de muñeca, para que sus usuarios puedan realizar actividades de la vida diaria.

Con esto, esperamos proponer a desarrolladores de este tipo de sistemas una arquitectura tan ligera como sea posible basándose en una sola característica; esto con el propósito de facilitar la migración de los modelos a los microcontroladores utilizados en estos tipos de dispositivos, Para ello, este artículo se organiza de la siguiente manera. La información sobre los datos se analiza en la Sección 2. La Sección 3 presenta la extracción de características de las señales electromiográficas. Los resultados y discusión se registran en la Sección 4. Y las conclusiones se abordan en la Sección 5.

# Materiales y Métodos

En este artículo se analizan datos de EMG de superficie de la base de datos NinaPro DB1. Las señales de EMG se tomaron usando diez electrodos activos, de doble diferencial, marca OttoBock MyoBock 13E200, a un intervalo constante de 100Hz utilizando una tarjeta National Instruments DAQ (NI-DAQ PCMCIA 6024E) con una resolución de 12 bits. El foco se pone en el grupo de ejercicios 3, agarres, también identificados como grupo C, tomando los datos para los 27 sujetos y las 10 repeticiones por ejercicio realizadas por cada uno, y en particular en los ejercicios 1 y 23 y la posición de reposo, 0; el ejercicio 1 corresponde a un agarre de gran diámetro y el 23 a un agarre de cuchillo con extensión del dedo índice (figura 1). Cada repetición de movimiento dura 5

segundos, seguida de 3 segundos de descanso.

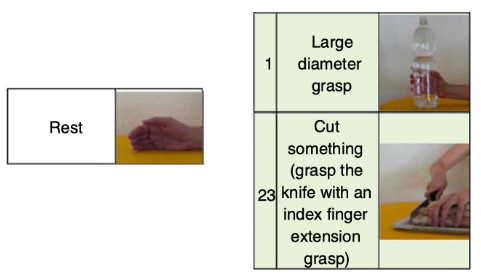


Figura 1. Tipos de agarres explorados.

En la figura 2, se puede visualizar un caso típico en el cual se sustraen las señales sEMG, el numero de la postura (1, 10 y 27) y el numero de la repetición (valor entre 1 y 10). Aquellos valores en los cuales tanto la postura como el numero de repetición corresponden a 0 están asociados a la posición de descanso.

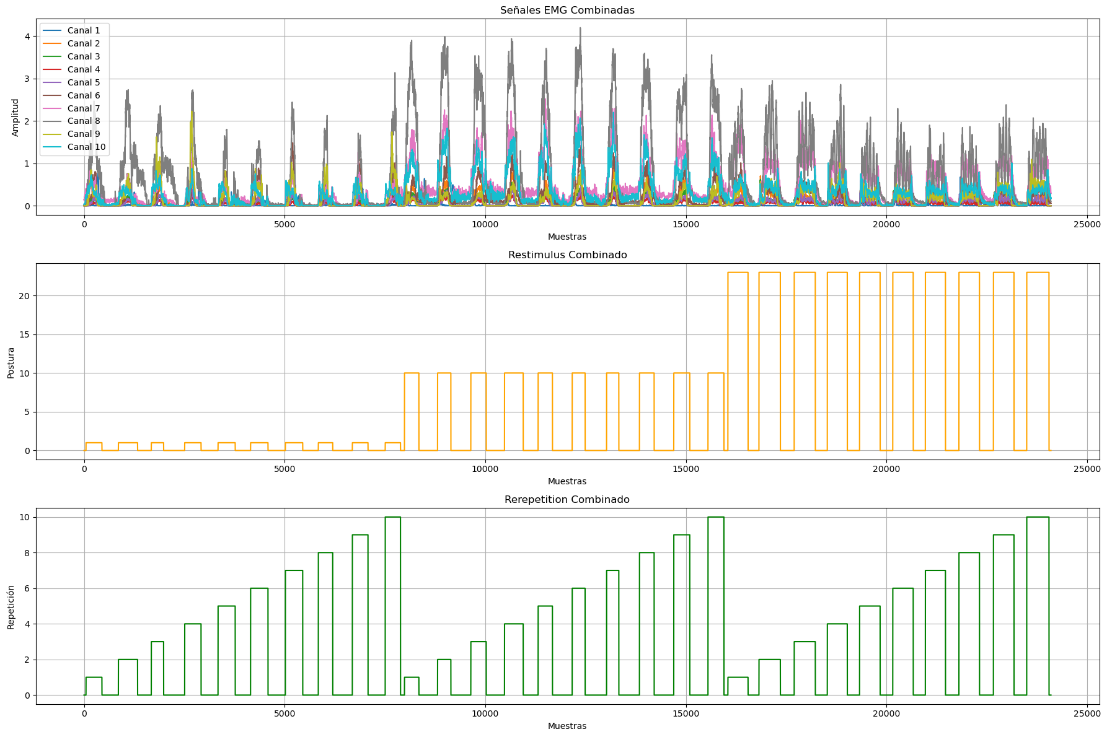


Figura 2. Señales de interés para un sujeto cualquiera. Arriba) Señal sEMG para 10 canales simultáneos. Medio) Numero de postura para cada una de las repeticiones, mencionado por los autores como “restimulus”. Abajo) Numero de repetición para cada uno de las posturas, mencionado por los autores como “rerepetition”.

Los datos se preprocesan y luego se extraen sus características usando RMS (del inglés, Root Mean Square), WL (del inglés, Waveform Length) e IAV (del inglés, Integrated Absolute Value). Es importante anotar, que todos los procedimientos previamente mencionados se realizaron empleando las funciones de la biblioteca LibEMG [13].

## Fuente de datos

La base de datos DB1 es de acceso público [13], se descarga como un conjunto de archivos ZIP, cada archivo ZIP corresponde a cada uno de los sujetos que participaron en el experimento, los cuales no presentaban amputación de ninguna extremidad: S1.zip, hasta S27.zip. Al descomprimir el archivo ZIP de un sujeto específico (por ejemplo, S1.zip), se obtienen varios archivos en formato .mat cada uno de estos archivos .mat corresponde a un ejercicio realizado por el sujeto.

Cada archivo .mat[[1]](#footnote-1) (por ejemplo, Sx\_A1\_Ey.mat, Sx\_A1\_Ey.mat, etc.) contiene variables específicas. Las variables más importantes y comunes dentro de cada archivo .mat son (ver Tabla 1):

* **emg**: es una matriz donde cada columna representa un canal de EMG (generalmente 10, correspondientes a los sensores sEMG). Las filas representan las muestras de EMG a lo largo del tiempo.
* **stimulus (o restimulus o gesture)**: es una columna vectorial que contiene la etiqueta del movimiento que se está realizando en cada instante de tiempo. Cero '0' reposo, y los números subsiguientes (1, 2, 3, ...) corresponden a los diferentes movimientos especificados.
* **subject**: escalar que indica el número del sujeto.
* **exercise**: escalar que indica el número del ejercicio (1, 2 o 3 para los tres tipos de ejercicios en DB1).
* **repetition**: vector que indica el número de repetición del gesto.
* **frequency**: frecuencia de muestreo de los datos EMG (100 Hz para DB1 después del preprocesamiento de Ninapro).

Tabla 1. Características dataset ejercicio 3 de los 27 sujetos.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Descripción** | **Tipo de dato** |
| subject | Número del sujeto | Categórico |
| exercise | Número del ejercicio3 | Categórico |
| emg (12 col) | Señales sEMG. | Numérico |
| glove (22 col) | Señales sin procesar de del guante Cyberglove II. | Numérico |
| stimulus (1 col) | Movimiento repetido por el sujeto al seguir las instrucciones. | Categórico |
| restimulus (1 col) | Etiqueta colocada a posteriori para el movimiento repetido por el sujeto. | Categórico |
| repetition (1 col) | Repetición de stimulus | Categórico |
| rerepetition (1 col) | Repetición de rstimulus | Categórico |

Después de descomprimir cada uno de los archivos .zip de cada sujeto, el siguiente paso consiste en agrupar en un mismo directorio todos los archivos correspondientes al ejercicio 3 (Sx\_Ay\_E3.mat). Una vez agrupados los archivos archivos .mat, se procede a generar, a partir de estos, los archivos .csv correspondientes a cada una de las características de la tabla 1. Finalmente, como solo nos interesan las características asociadas a las señales sEMG (subject, emg, rerepetition y restimulus) se seleccionan aquellos archivos .csv que tengan que ver con estas y se ponen en un directorio aparte.

Luego se procedió combinar cada uno de estos archivos en una única matriz numpy que se almacena como un binario (extensión “.npy”). Cada uno de estos archivos contiene todos los datos sEMG crudos de cada sujeto para cada una de las posturas analizadas por Ninapro. En la figura 3 se ilustra una de las señales asociadas al sujeto 1.

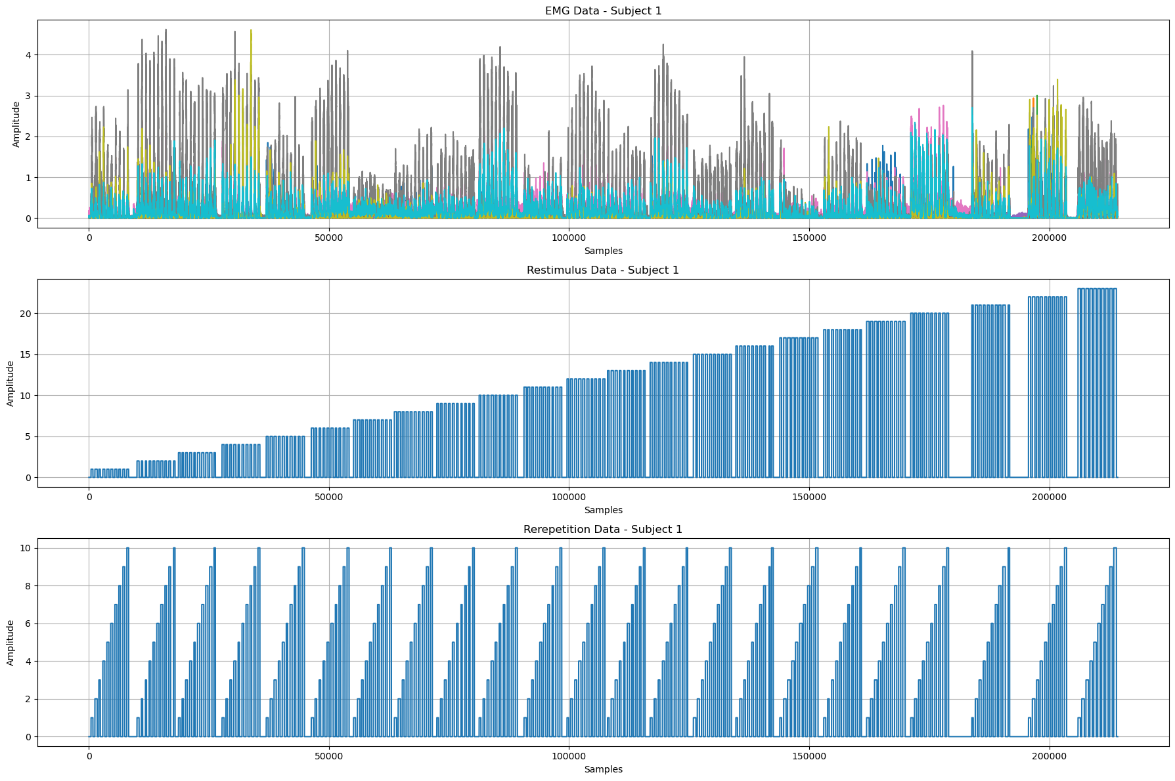


Figura 3. Muestra señales sEMG sujeto 1

Como solo interesan las posturas de los ejercicios 1 y 23, incluyendo los descansos, se procede a seleccionar las muestras correspondientes, ver figura 3, y se guardan en otro archivo .npy para luego realizar sobre estas el procesamiento digital de señales para filtrado, enventanado y extracción de características.

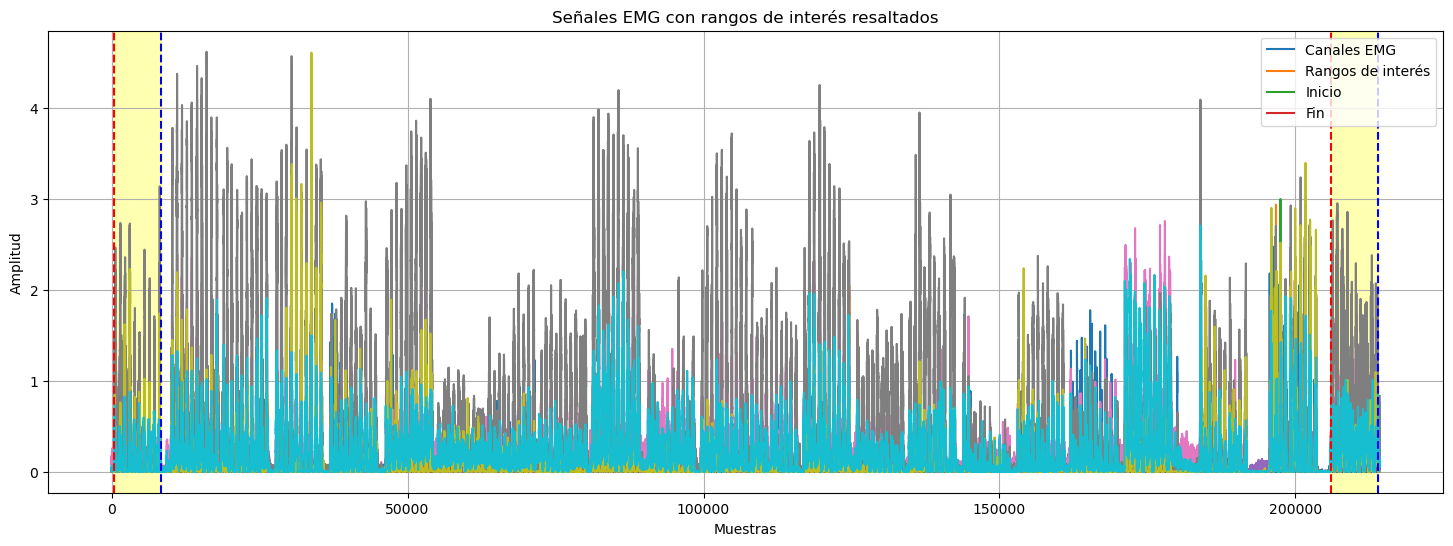


Figura 4. Posturas seleccionadas

## Procesamiento de las señales musculares

El procesamiento de los datos se realiza en 2 fases, como se ilustra en la figura 4: preprocesamiento o preparación de los datos y extracción de características.

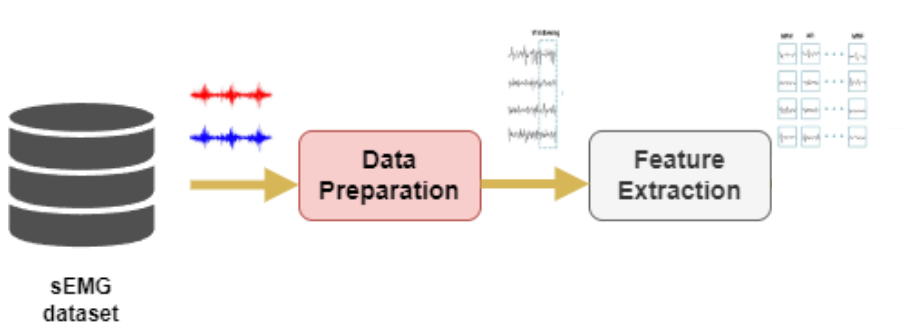


Figura 5. Método de preprocesamiento y extracción de características.

Para este caso el preprocesamiento se refiere a la obtención de los datos crudos de las posturas elegidas (1, 23 y descanso) desde los archivos .npy y la extracción de estos hasta convertirlos en el dataset que será empleado como entrada para el entrenamiento y validación de los modelos de ML y DL. En la figura 5 se ilustran los datos crudos correspondientes a las posturas 1 y 23 (incluyendo descansos) para el sujeto 1.

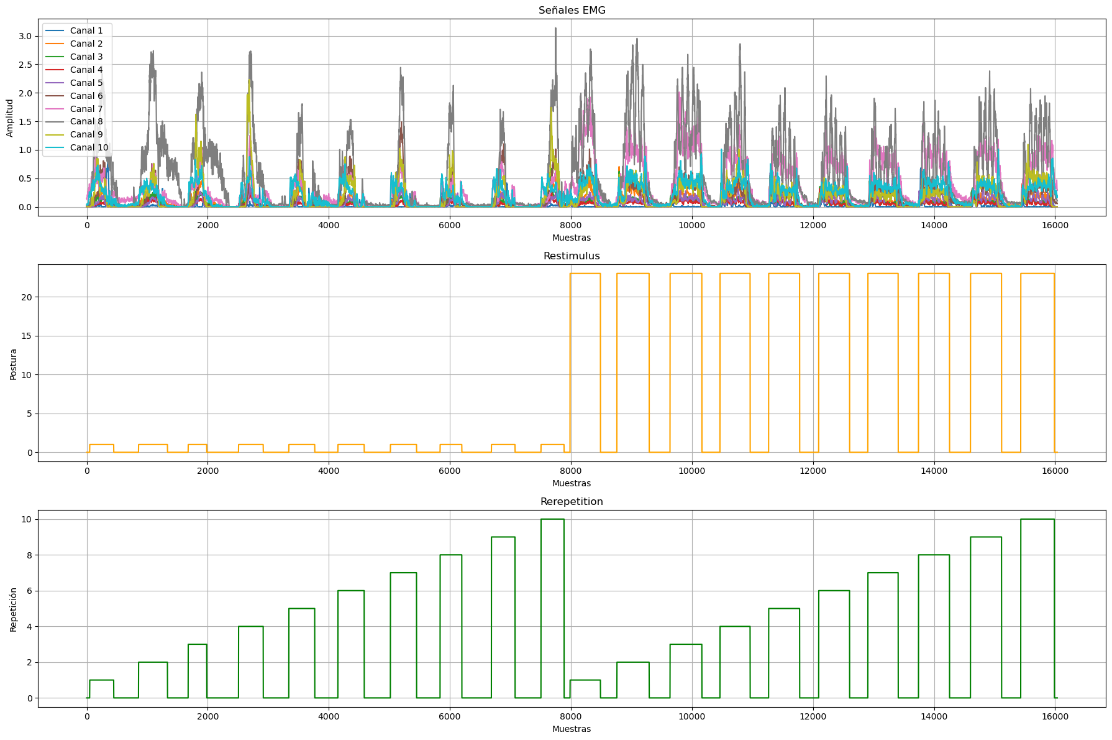


Figura 6. Datos crudos del sujeto 1 para las posturas de interés.

Como la señal es muy ruidosa lo primero que se debe realizar es filtrarla, para el caso se emplea un filtro pasa bajas digital cuyas frecuencias de corte y muestreo son respectivamente 1 Hz y 100 Hz. El resultado de este procedimiento es una señal más limpia y lista para la extracción de características (figura 6):

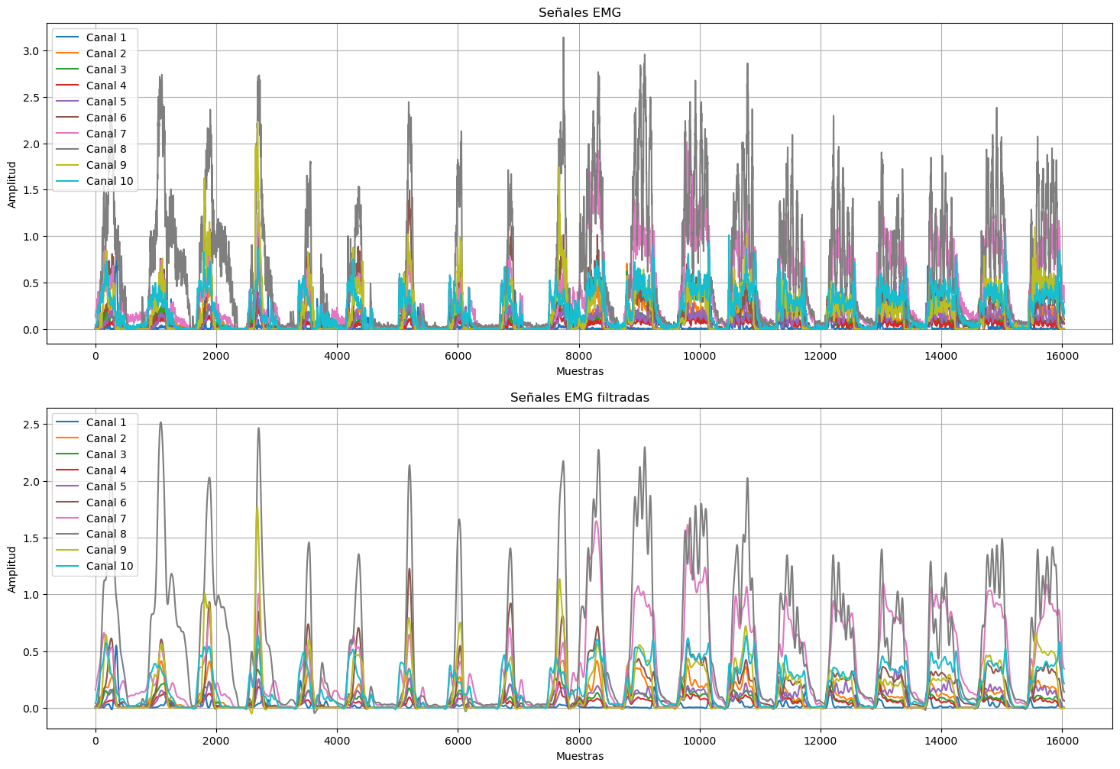


Figura 7. Señales sEMG crudas y filtradas

Luego se llevó a cabo un proceso de segmentación sobre la señal filtrada. A partir de cada segmento o ventana de señal seleccionada se calcularon las características elegidas: RMS, WL, e IAV.

Se emplearon ventanas de 30 muestras (equivalentes a 300 ms), con un solapamiento de 10 muestras (100 ms) entre ventanas consecutivas. Una vez calculadas las características para todas las ventanas se generó un archivo .csv por cada sujeto; en este se almacenaron las características extraídas junto con las etiquetas (labels) correspondientes a la postura asociada a cada ventana.

Cabe destacar que además de las etiquetas asociadas a las posturas de interés (0, 1 y 23), se introdujo una etiqueta adicional, identificada con el valor -1, para aquellas ventanas que contenían muestras correspondientes a transiciones entre el estado de reposo y una postura, o viceversa. Este nuevo label tuvo como finalidad señalar que la ventana incluía una fase de transición y no debía considerarse como una postura estable.

## Modelos de inteligencia artificial

Las características extraídas en la fase anterior se utilizaron como entrada para los modelos de clasificación. Se evaluaron diferentes modelos ML y DL para clasificar las tres (3) clases de posturas seleccionadas, las posturas 1, 23, y estado de reposo (0), pertenecientes al conjunto de ejercicios 3 de la base de datos Ninapro DB1. En esta configuración se optó por una estrategia de modelado independiente del sujeto, considerando que los datos entre individuos presentan una alta variabilidad y una baja correlación.

A partir de cada archivo .csv generado por sujeto se construyeron tres matrices distintas, una por cada característica extraída (RMS, WL e IAV), acompañadas de sus respectivas etiquetas (labels), que incluyeron las clases 0, 1, 23 y -1. Esta última clase, como se señaló anteriormente, representa ventanas asociadas a transiciones.

Para cada característica se generaron 10 columnas correspondientes a los 10 canales de adquisición de la señal EMG. Por ejemplo, en el caso de RMS, las columnas se nombraron como RMS\_ch1, RMS\_ch2, ..., RMS\_ch10. De forma análoga se nombraron las columnas para WL (WL\_ch1 a WL\_ch10) e IAV (IAV\_ch1 a IAV\_ch10). A continuación, en la tabla 2,se muestra la estructura general de estas matrices:

Tabla 2. Dataset de características de las señales sEMG para las posturas elegidas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Matriz** | **Variable** | **Descripción** | **Tipo de dato** |
| RMS | RMS\_chx[[2]](#footnote-2) (10 col) | Característica RMS | Numérico |
| label | Postura (0, 1, 23, -1) | Categórico |
| WL | WL\_chx (10 col) | Característica WL | Numérico |
| label | Postura (0, 1, 23, -1) | Categórico |
| IAV | IAV\_chx (10 col) | Característica IAV | Numérico |
| label | Postura (0, 1, 23, -1) | Categórico |

Dado que los datos de entrada incluían cuatro etiquetas distintas (0, 1, 23 y -1) el primer paso consistió en eliminar las entradas correspondientes a ventanas con la etiqueta -1, ya que no correspondían a un único agarre sino una postura de transición que no podría ser debidamente etiquetada. De este modo el conjunto de datos resultante para el entrenamiento y validación de los modelos quedó únicamente con las clases de interés: 0, 1 y 23.

Es importante resaltar que debido a que el estado de reposo es más común en las bases de datos analizada las clases quedaron desbalanceadas en los dataset, con una mayoría de muestras correspondientes al estado de reposo, clase 0; por lo tanto, se requirió aplicar estrategias diferenciadas según el tipo de modelo utilizado. El numero total de ventanas por postura se muestra en la figura X.

VENTANAS POR POSTURA

Para cada característica (RMS, WL, IAV), el conjunto de datos fue dividido en 80% para entrenamiento y un 20% para validación. Para los modelos de aprendizaje automático se aplicó un proceso de submuestreo aleatorio (undersampling) para balancear las clases durante el entrenamiento. Por otro lado, para el modelo de aprendizaje profundo se utilizó un muestreo estratificado, con el fin de conservar la proporción original de clases tanto en el conjunto de entrenamiento como en el de prueba.

Además, antes de llevar los datos a los modelos de ML y DL, se aplicó un procedimiento de escalado de las características seleccionadas para normalizar los datos y facilitar la convergencia durante el entrenamiento.

La siguiente tabla resume los modelos que fueron considerados en el proceso de evaluación:

Tabla 3. Resumen de las características de los modelos probados.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Categoría** | **Tipo** | **Modelo** | **Entradas** | **Balanceo** | **Salidas** |
| Aprendizaje de máquina | SVM | 1 | RMS | Si | 0, 1, 23 |
| 2 | WL | Si | 0, 1, 23 |
| 3 | IAV | Si | 0, 1, 23 |
| KNN | 4 | RMS | Si | 0, 1, 23 |
| 5 | WL | Si | 0, 1, 23 |
| 6 | IAV | Si | 0, 1, 23 |
| RF | 7 | RMS | Si | 0, 1, 23 |
| 8 | WL | Si | 0, 1, 23 |
| 9 | IAV | Si | 0, 1, 23 |
| Aprendizaje profundo | CNN-LTSM | 10 | RMS | No | 0 [0], 1 [1], 2 [23] |
| 11 | WL | No | 0 [0], 1 [1], 2 [23] |
| 12 | IAV | No | 0 [0], 1 [1], 2 [23] |

Como se puede observar, en el caso del modelo de aprendizaje profundo, las salidas presentan una codificación diferente ya que se realizó un procedimiento de **relabeling**, en el cual las etiquetas originales 0, 1 y 23 (valores entre corchetes) fueron mapeadas a los valores 0, 1 y 2, respectivamente. Esta transformación facilitó el manejo de las clases en la función de salida (output layer) del modelo que se diseñó para realizar clasificación multiclase con codificación categórica.

Los parámetros específicos para cada uno de los modelos implementados se describen a continuación:

* **Máquinas de vectores de soporte (SVM)**: clasificador eficaz en espacios de alta dimensión, sensible a datos desbalanceados.

Tabla 4. Parámetros para SVM

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Entradas** | **Características específicas** |
| 1 | RMS | * **kernel**: RBF * **C**: 100 * **gamma**:1 |
| 2 | WL | * **kernel**: RBF * **C**: 100 * **gamma**:0.1 |
| 3 | IAV | * **kernel**: RBF * **C**: 100 * **gamma**:1 |

* **K-Nearest Neighbors (KNN)**: clasificación basada en distancia.

Tabla 5. Parámetros para KNN

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Entradas** | **Características específicas** |
| 3 | RMS | * **Vecinos**: 3 * **Métrica**: Manhattan |
| 4 | WL | * **Vecinos**: 9 * **Métrica**: Manhattan |
| 5 | IAV | * **Vecinos**: 3 * **Métrica**: Manhattan |

* **Random Forest (RF)**: ensamble de árboles de decisión.

Tabla 6. Parámetros para RF

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Entradas** | **Características específicas** |
| 6 | RMS | * **Número de árboles (n\_estimators)**: 200 * **Profundidad máxima de los árboles (max\_depth)**: 20 * **Mínimo número de muestras por hoja (min\_samples\_leaf)**: 1 * **Mínimo número de muestras para dividir un nodo (min\_samples\_split)**: 2 * **Criterio de división**: Gini (valor por defecto) |
| 7 | WL | * **Número de árboles (n\_estimators)**: 200 * **Profundidad máxima de los árboles (max\_depth)**: None (sin límite) * **Mínimo número de muestras por hoja (min\_samples\_leaf)**: 1 * **Mínimo número de muestras para dividir un nodo (min\_samples\_split)**: 2 * **Criterio de división**: Gini (valor por defecto) |
| 8 | IAV | * **Número de árboles (n\_estimators)**: 200 * **Profundidad máxima de los árboles (max\_depth)**: 20 * **Mínimo número de muestras por hoja (min\_samples\_leaf)**: 1 * **Mínimo número de muestras para dividir un nodo (min\_samples\_split)**: 2 * **Criterio de división**: Gini (valor por defecto) |

* **Arquitectura hibrida CNN-LTSM**: arquitectura híbrida que combina una red convolucional 1D para extraer patrones locales de las señales EMG, seguida de una capa LSTM que modela la dependencia temporal entre ventanas. Esta combinación permite capturar tanto la estructura espacial como la secuencialidad de los datos. La arquitectura empleada en nuestro caso fue propuesta por Lin Chen et al [14], se muestra en la siguiente figura:

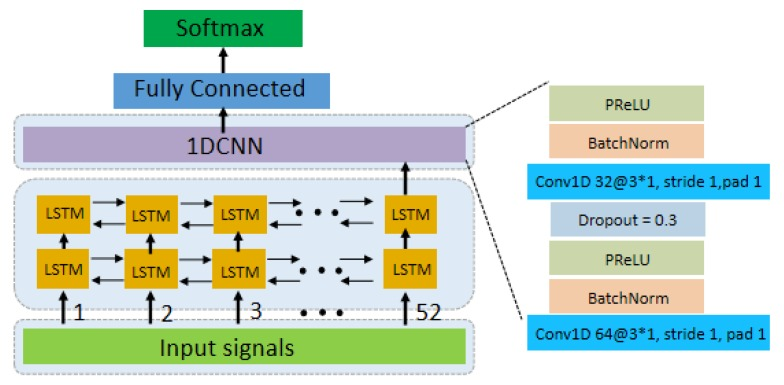


Figura 8. Arquitectura red CNN-LSTM propuesta por Lin Chen et Al [14]

La tabla 4 resume la arquitectura de red previamente descrita y que usa los mismos parámetros de entrada de los modelos RMS (Modelo 10), WF (Modelo 11) e IAV (Modelo 12):

Tabla 7. Arquitectura de red para CNN-LTSM

|  |  |  |
| --- | --- | --- |
| **Capa (Tipo)** | **Forma de salida** | **Parámetros** |
| input\_signals (Capa de entrada) | (Ninguno, 1, 10) | 0 |
| lstm\_1 (LSTM) | (Ninguno, 1, 64) | 19200 |
| lstm\_2 (LSTM) | (Ninguno, 1, 64) | 33024 |
| conv1d\_1 (Conv1D) | (Ninguno, 1, 64) | 12352 |
| batchnorm\_1 (Normalización por lotes) | (Ninguno, 1, 64) | 256 |
| prelu\_1 (PReLU) | (Ninguno, 1, 64) | 64 |
| dropout\_1 (Dropout) | (Ninguno, 1, 64) | 0 |
| conv1d\_2 (Conv1D) | (Ninguno, 1, 32) | 6176 |
| batchnorm\_2 (Normalización por lotes) | (Ninguno, 1, 32) | 128 |
| prelu\_2 (PReLU) | (Ninguno, 1, 32) | 32 |
| global\_avg\_pooling (Agrupación promedio global 1D) | (Ninguno, 1, 32) | 0 |
| fully\_connected\_dense (Densa) | (Ninguno, 1, 32) | 1056 |
| prelu\_fc (PReLU) | (Ninguno, 1, 32) | 32 |
| output\_softmax (Densa) | (Ninguno, 3) | 99 |

## Métricas de evaluación

El rendimiento de los modelos de clasificación se midió utilizando las siguientes métricas:

* **Precisión:** proporción de predicciones positivas correctas:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Donde,

* + **TP**: verdaderos positivos (correctamente clasificados como positivos).
  + **FP**: falsos positivos (incorrectamente clasificados como positivos).
* **Recall**: proporción de casos positivos reales que fueron correctamente identificados:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Donde,

* + **FN**: falsos negativos (casos positivos incorrectamente clasificados como negativos).
  + **TP**: verdaderos positivos (correctamente clasificados como positivos).

Para ambos casos se considera porcentajes superiores al 80% para la selección de los modelos.

* **F1**-**score**: Media armónica de las métricas de precisión y recall:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

* **ROC** (Del inglés, Receiver Operating Characteristic): gráfico de Tasa de Verdaderos Positivos (Recall) vs. Tasa de Falsos Positivos (FP/(FP+TN)) para distintos umbrales de clasificación.
* **AUC** (Área bajo la curva ROC): medida de la capacidad del clasificador para distinguir entre clases. El valor de AUC se encuentra entre 0 y 1 (1 = clasificador perfecto, 0.5 = aleatorio).

# Extracción de características

RMS, WL e IAV son métodos comunes para la extracción de características en señales de electromiografía superficial (sEMG). Su objetivo es reducir la dimensionalidad de la señal y capturar información relevante para aplicaciones como control de prótesis, diagnóstico médico o interfaz cerebro-máquina.

* **RMS** (Root Mean Square): representa la energía promedio de la señal sEMG en una ventana de tiempo. Es útil para estimar la activación muscular (mayor RMS = mayor contracción). Presenta robustez frente a ruido (suaviza variaciones rápidas). Útil para diferenciar entre reposo y contracción.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Donde,

* + **xi**: Muestra de la señal sEMG en el instante i.
  + **N**: Número de muestras en la ventana.
* **WL** (Waveform Length): mide la complejidad temporal de la señal (suma de diferencias entre muestras consecutivas). Es sensible a cambios en la amplitud y frecuencia de la sEMG. Indicador de patrones de activación muscular. Ayuda a distinguir gestos rápidos (ej: pinza) vs. lentos (ej: agarre sostenido).

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Donde,

* + **xi**: Muestra de la señal sEMG en el instante i.
  + **N**: Número de muestras en la ventana.

Tiene aplicación en discriminación de gestos (ej: mano abierta vs. puño), detección de movimientos dinámicos.

* **IAV** (Integral Absolute Value): calcula el área bajo la curva de la señal sEMG en valor absoluto. Relacionado con la actividad muscular total en un intervalo. Similar a RMS pero menos sensible a picos abruptos.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Donde,

* + **xi**: Muestra de la señal sEMG en el instante i.
  + **N**: Número de muestras en la ventana.

Su aplicación en sEMG es útil para estimación de la intensidad de la contracción, cuantifica la fuerza aplicada. Usado ampliamente en sistemas de control en tiempo real por su bajo costo computacional.

# Resultados y Discusión

En cuanto a los modelos de aprendizaje supervisado, la optimización de hiperparámetros se llevó a cabo utilizando la técnica *grid search* que implicó la exploración exhaustiva de los parámetros definidos en la siguiente tabla 8.

Tabla 8. Hiperparámetros a optimizar

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Hiperparametro** | **Valores** |
| SVM | C (Regularización) | 0.1, 1, 10, 100 |
| Kernel | 'rbf', 'linear' |
| Gamma | 1, 0.1, 0.01, 0.001 |
| KNN | Número de vecinos | 3, 5, 7, 9 |
| Función de peso | 'uniform', 'distance' |
| Algoritmo para calcular vecinos | 'auto', 'ball\_tree', 'kd\_tree' |
| Metric | 'euclidean', 'manhattan', 'minkowski' |
| RF | Número arboles | 50, 100, 200 |
| Profundidad máxima | None, 10, 20 |
| Número de muestras mínimo para división de nodo interno | 2, 5, 10 |
| Número de muestras mínimo de un nodo rama | 1, 2, 4 |

En este proceso se generaron 160 combinaciones para los hiperparámetros y se seleccionaron los modelos para SVM, KNN y RF cuya combinación arrojó la mejor precisión (accuracy) y cuyos resultados se reúnen en la tabla 9.

Tabla 9. Resultados de los modelos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelo** | **Características** | **Precisión (M/W)** | **Recall (M/W)** | **F1-Score (M/W)** | **AUC** |
| 1 - SVM(RMS) | * **kernel**: RBF * **C**: 100 * **gamma**:1 | 0.95 | 0.95 | 0.95 | * **0:** 0.99 * **1:** 0.98 * **23:** 0.99 |
| 2 - SVM(WL) | * **kernel**: RBF * **C**: 100 * **gamma**:0.1 | 0.83 | 0.83 | 0.83 | * **0:** 0.96 * **1:** 0.93 * **32:** 0.94 |
| 3 - SVM(IAV) | * **kernel**: RBF * **C**: 100 * **gamma**:1 | 0.94 | 0.94 | 0.94 | * **0:** 0.98 * **1:** 0.98 * **23:** 0.99 |
| 4 - KNN(RMS) | * **Vecinos**: 3 * **Métrica**: Manhattan | 0.95 | 0.95 | 0.95 | * **0:** 0.98 * **1:** 0.98 * **23:** 0.99 |
| 5 - KNN(WL) | * **Vecinos**: 9 * **Métrica**: Manhattan | 0.84 | 0.84 | 0.84 | * **0:** 0.96 * **1:** 0.94 * **23:** 0.95 |
| 6 - KNN(IAV) | * **Vecinos**: 3 * **Métrica**: Manhattan | 0.95 | 0.95 | 0.95 | * **0:** 0.98 * **1:** 0.98 * **23:** 0.99 |
| 7 - RF(RMS) | * **Número de árboles**: 200 * **Profundidad máxima de los árboles**: 20 * **Mínimo número de muestras por hoja**: 1 * **Mínimo número de muestras para dividir un nodo**: 2 | 0.94 | 0.94 | 0.94 | * **0:** 0.99 * **1:** 0.99 * **2:** 0.99 |
| 8 - RF(WL) | * **Número de árboles**: 200 * **Profundidad máxima de los árboles**: None * **Mínimo número de muestras por hoja**: 1 * **Mínimo número de muestras para dividir un nodo**: 2 | 0.85 | 0.85 | 0.85 | * **0:** 0.97 * **1:** 0.95 * **23:** 0.95 |
| 9 - RF(IAV) | * **Número de árboles**: 200 * **Profundidad máxima de los árboles**: 20 * **Mínimo número de muestras por hoja**: 1 * **Mínimo número de muestras para dividir un nodo**: 2 | 0.94 | 0.94 | 0.94 | * **0:** 0.94 * **1:** 0.94 * **23:** 0.94 |
| 10 - CNN-LSTM(RMS) | * **Conv1D** (×2) * **LSTM** (×2, 52 celdas, 64 unidades) * **Capa de salida** | 0.94 / 0.93 | 0.92 / 0.93 | 0.92 / 0.93 | * **0 [0]:** 0.99 * **1 [1]:**  0.98 * **2 [23]:** 0.99 |
| 11 - CNN-LSTM(WL) | 0.82 / 0.84 | 0.81 / 0.84 | 0.82 / 0.84 | * **0 [0]:** 0.96 * **1 [1]:** 0.94 * **2 [23]:** 0.94 |
| 12 - CNN-LSTM(IAV) | 0.93 / 0.93 | 0.91 / 0.93 | 0.92 / 0.93 | * **0 [0]:** 0.99 * **1 [1]:**  0.98 * **2 [23]:** 0.99 |

También se mantuvieron los 10 canales de adquisición de las señales para el entrenamiento y validación de los modelos debido a que presentan una baja correlación entre sí según se aprecia en la figura 9 para RMS, figura 10 para WL y figura 11 para IAV:

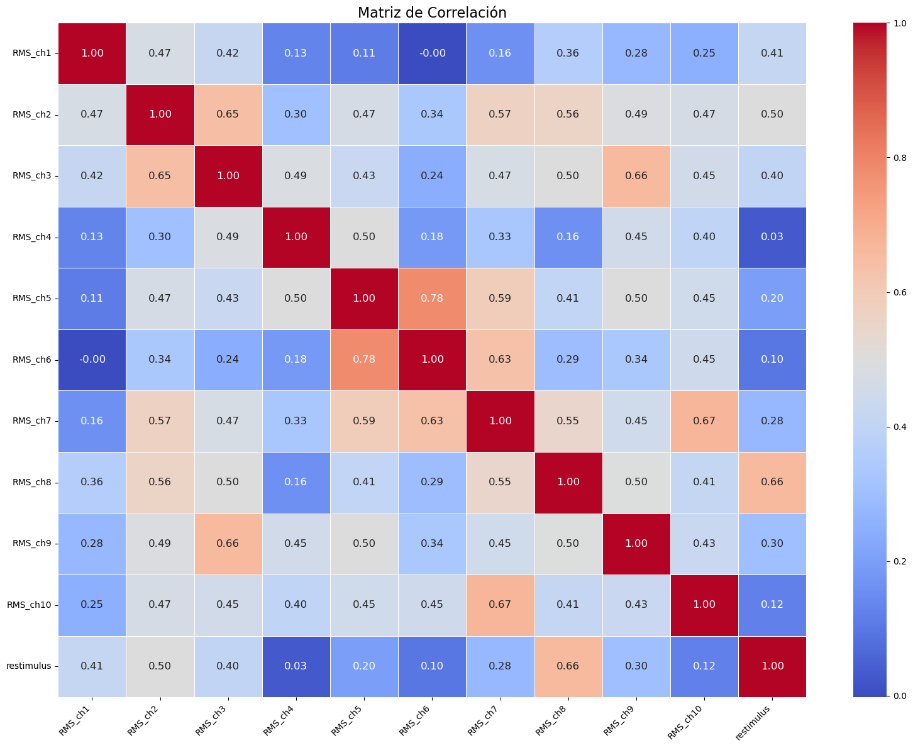


Figura 9. Matriz de correlación para RMS

A blue and red squares with white text

AI-generated content may be incorrect.

Figura 10. Matriz de correlación para WL

A colorful squares with red and blue squares

AI-generated content may be incorrect.

Figura 11. Matriz de correlación para IAV

SOBRE LOS RESULTADOS DE CORRELACIóN EN SVM/IVA > WL

Los mismos resultados de la tabla 8 podemos verlos representados en las respectivas matrices de confusión para SMV, figura 12, WL, figura 13 e IAV, figura 14 puestas a continuación:

|  |  |  |
| --- | --- | --- |
| A blue squares with numbers  AI-generated content may be incorrect. | A blue squares with numbers  AI-generated content may be incorrect. | A graph with numbers and squares  AI-generated content may be incorrect. |

Figura 12. Matrices de confusión RMS, WL e IAV para el modelo SVM

|  |  |  |
| --- | --- | --- |
| A blue squares with numbers  AI-generated content may be incorrect. | A blue squares with numbers  AI-generated content may be incorrect. | A graph with blue squares  AI-generated content may be incorrect. |

Figura 13. Matrices de confusión RMS, WL e IAV para el modelo KNN

|  |  |  |
| --- | --- | --- |
| A graph with blue squares  AI-generated content may be incorrect. | A blue squares with numbers  AI-generated content may be incorrect. | A graph with numbers and squares  AI-generated content may be incorrect. |

Figura 14. Matrices de confusión RMS, WL e IAV para el modelo RF

Las curvas ROC también nos ayudan a apreciar los resultados de los modelos y como era de esperarse, estas son muy similares para RMS e IAV debido a la relativa equivalencia existente entre estas 2 características extraídas de las señales según se indicó antes y a que estas son especialmente útiles para detectar su energía según se aprecia en las figuras 15, 16 y 17. De igual forma se observa en las figuras que WF tiene unos resultados inferiores ya que esta es especialmente útil para detectar cambios en la amplitud y frecuencia de la sEMG. ayudando a distinguir gestos rápidos y estos no son tan apreciables por la misma naturaleza de los ejercicios propuestos en la base de datos DB1.

|  |  |  |
| --- | --- | --- |
| A graph of a function  AI-generated content may be incorrect. | A graph of a function  AI-generated content may be incorrect. | A graph of a function  AI-generated content may be incorrect. |

Figura 15. Curvas ROC de RMS, WL e IAV para el modelo SVM

Las curvas ROC presentan una clasificación cercana a un clasificador perfecto, por lo tanto, no son sensibles a la selección de umbral; siendo WL la característica mas sensible entre las tratadas. SOBRE LA FORMA DE LA CURVA ROC EN WL: la postura 0 puede funcionar bien con un umbral relativamente mas bajo que las otras dos, puesto que rápidamente sube casi hasta 1; mientras que, la postura 23 requiere umbrales mas bajos para aumentar la sensibilidad de esta clase al compararse con la clase 1; aunque podría repercutir en una sobre clasificación para esta clase.

|  |  |  |
| --- | --- | --- |
| A graph of a line  AI-generated content may be incorrect. | A graph of a line graph  AI-generated content may be incorrect. | A graph of a function  AI-generated content may be incorrect. |

Figura 16. Curvas ROC de RMS, WL e IAV para el modelo KNN

|  |  |  |
| --- | --- | --- |
| A graph of a function  AI-generated content may be incorrect. | A graph of a function  AI-generated content may be incorrect. | A graph of a function  AI-generated content may be incorrect. |

Figura 17.Curvas ROC de RMS, WL e IAV para el modelo RF

Las matrices de confusión y las curvas ROC anteriores están relacionadas, como se indica en las gráficas asociadas, a los modelos de ML, en cuanto al modelo de DL implementado los resultados de la matriz de confusión de aprecian en la figura 18 y la curva ROC en la figura 19. Es importante recordar que para este modelo los datos no fueron balanceados y que se realizó un relabeling, indicado en la columna AUC de la tabla 9 con los corchetes cuadrados.

|  |  |  |
| --- | --- | --- |
| **A blue and white graph  AI-generated content may be incorrect.** | **A graph with numbers and labels  AI-generated content may be incorrect.** | **A blue squares with numbers and labels  AI-generated content may be incorrect.** |

Figura 18. Matrices de confusión RMS, WL e IAV para el modelo CNN-LTSM

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figura 19.Curvas ROC de RMS, WL e IAV para el modelo CNN-LTSM

En este estudio se observan resultados muy similares entre los diferentes modelos de ML implementados, los mejores valores, por encima del 94% en el f1-score se tienen para RMS e IAV, situación que era de esperarse por la naturaleza de las señales de la base de datos analizada (movimientos de 5 segundos por ejercicio). WL, que es útil para detectar movimientos rápidos en las señales también arroja resultados destacables con valores superiores al 83% para f1-score. En cuanto a DL los resultados son muy similares a ML con valores para f1-score de 92% en entrenamiento y 93% en validación para RMS e IAV, y para WL f1-score de 82% y 84% respectivamente.

Queda para estudios futuros seguir evaluando la validez de los modelos de ML implementándolos con un mayor número de posiciones de mano, agarres para nuestro caso, llevándolos de los 2 implementados acá a 3 o más y combinando también característica extraídas de las señales formando, por ejemplo, un data set con los datos de RMS y WL.

# Conclusiones

Los modelos de ML siguen estando vigentes para el desarrollo de controles de prótesis debido a su simplicidad y a que pueden ser implementados en los microprocesadores involucrados en este tipo de soluciones que, generalmente, tienen limitaciones en memoria.

El desarrollo de prótesis de miembro superior usando señales de electromiografía es un campo de estudio vigente, buscando mejorar la calidad de vida de quienes presentan perdidas de este tipo, con este fin se han dispuesto base de datos públicas con registros de señales para que sean usadas por los investigadores y desarrolladores. Por otro lado, los avances en modelos de inteligencia artificial han dinamizado este campo. Una de las principales dificultades está en los modelos de aprendizaje automático y como llevarlos a los sistemas de control de este tipo de elementos dadas las limitaciones, principalmente de memoria, de los microprocesadores disponibles. En ese orden de ideas evaluar modelos simples de ML, con 2 o tres tipos de posiciones de mano y una característica extraída de las señales es completamente pertinente.

En análisis de señales de electromiografía se aprecia unos mejores resultados para características de las señales que involucran su energía, como RMS e IAV. A pesar de que en la actualidad se ha buscado implementar técnicas de Deep Learning como las CNN + LSTM que se presentan en este artículo, se demuestra mediante el presente estudio que una selección de características de energía, luego de un procesamiento adecuado de la señal (rectificado, filtrado, entre otras) obtienen resultados comparables para pocos agarres como es usual encontrar en prótesis comerciales.

El uso de técnicas de Deep Learning por lo general requiere de grandes volúmenes de datos para evitar un overfitting a los usuarios, agarres o condiciones de entrenamiento; por otra parte, la extracción de características convencional, en particular estas de energía, permite una interpretabilidad de los modelos y posible generalización con base en las transformaciones realizadas sobre la señal. Cabe notar que los modelos de DL de este artículo fueron entrenados sobre la BD original previo a la realización del balanceo; el efecto de esta decisión es dificil de predecir, mas fue tomada para brindar al modelo la mayor cantidad de datos posibles. A futuro, se deberá evaluar sobre un dataset exactamente igual para poder comparar sobre la capacidad de clasificar posturas por las técnicas tradicionales de ML versus aquellas de DL.

# Referencias

[1] A. C. Roșca, C. C. Baciu, V. Burtăverde, and A. Mateizer, “Psychological Consequences in Patients With Amputation of a Limb. An Interpretative-Phenomenological Analysis,” *Front Psychol*, vol. 12, 2021, doi: 10.3389/fpsyg.2021.537493.

[2] Ottobock, “Myoelectric Hand System 8E70,” 2025. Accessed: Apr. 13, 2025. [Online]. Available: https://www.ottobock.com/en-us/product/8E70

[3] Open Bionics, “Hero Arm Overview,” Open Bionics. Accessed: Apr. 13, 2025. [Online]. Available: https://openbionics.com/en/hero-arm-overview/

[4] Prótesis Avanzadas Colombia, “Prótesis A3D,” Prótesis Avanzadas Colombia. Accessed: Apr. 13, 2025. [Online]. Available: https://www.protesisavanzadas.co/pr%C3%B3tesis-a3d

[5] exiii Inc., “Hackberry - Open-source 3D printable bionic hand,” exiii Inc. Accessed: Apr. 13, 2025. [Online]. Available: https://www.exiii-hackberry.com/

[6] OpenBionics, “OpenBionics – Open-source robotic and bionic devices,” OpenBionics. Accessed: Apr. 13, 2025. [Online]. Available: https://openbionics.org/

[7] Enabling The Future, “Enabling The Future – A Global Network Of Passionate Volunteers Using 3D Printing To Give The World A Helping Hand,” Enabling The Future. Accessed: Apr. 13, 2025. [Online]. Available: https://enablingthefuture.org/

[8] M. Suárez García, “Adaptación de la Prótesis de Mano Basada en Soft-Robotics PrExHand para la Evaluación con Usuarios no Patológicos en Pruebas Funcionales,” Escuela Colombiana de Ingeniería Julio Garavito, 2021. doi: 10.48713/10336\_31566.

[9] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori, “Comparison of six electromyography acquisition setups on hand movement classification tasks,” *PLoS One*, vol. 12, no. 10, p. e0186132, Oct. 2017, doi: 10.1371/journal.pone.0186132.

[10] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, “Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation,” *Sensors*, vol. 17, no. 3, 2017, doi: 10.3390/s17030458.

[11] J. Yang, M. Soh, V. Lieu, D. J. Weber, and Z. Erickson, “EMGBench: Benchmarking Out-of-Distribution Generalization and Adaptation for Electromyography,” 2024. [Online]. Available: https://arxiv.org/abs/2410.23625

[12] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, and T. G. Chang, “Surface Electromyography Signal Processing and Classification Techniques,” *Sensors*, vol. 13, no. 9, pp. 12431–12466, 2013, doi: 10.3390/s130912431.

[13] Ninapro Project Team, “DB1 Guidelines,” Ninapro Project – HEVS. Accessed: Apr. 13, 2025. [Online]. Available: https://ninapro.hevs.ch/instructions/DB1.html

1. Cada archivo .mat tiene el siguiente formato: Sx\_A1\_Ey.mat; donde x es el número del sujeto (1 hasta 27) e y corresponde al número del ejercicio (1, 2 o 3). [↑](#footnote-ref-1)
2. En RMS\_chx, la letra x está asociada al canal y toma valores entre 1 y 10 incluidos. [↑](#footnote-ref-2)