

2025

Desarrollo de algoritmos de aprendizaje profundo
para clasificación de movimientos de brazo mediante
señales EMG

Flavio Alfonso Juárez Castro



Universidad Autónoma de
Querétaro
Facultad de ingeniería

Desarrollo de algoritmos de
aprendizaje profundo para
clasificación de movimientos de brazo
mediante señales EMG

Tesis

Que como parte de los requisitos para obtener el
Grado de

Maestro en Ciencias en Inteligencia Artificial

Presenta

Flavio Alfonso Juárez Castro

Dirigido por:
Dr. Marco Antonio Aceves Fernández

Querétaro, Qro. a junio de 2025

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia
Artificial

**Desarrollo de algoritmos de aprendizaje profundo
para clasificación de movimientos de brazo mediante
señales EMG**

Tesis

Que como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta

Flavio Alfonso Juárez Castro

Dirigido por

Dr. Marco Antonio Aceves Fernández

Dr. Marco Antonio Aceves Fernández
Presidente

Dr. Efrén Gorrostieta Hurtado
Secretario

Dr. Juan Manuel Ramos Arreguín
Vocal

Dr. Saúl Tovar Arriaga
Suplente

Dr. Jesús Carlos Pedraza Ortega
Suplente

Centro Universitario, Querétaro, Qro. México
26 de junio del 2025

“Todos el esfuerzo es inútil si no crees en ti mismo”

— **Might Guy**

A mi familia: a mi padre, madre, hermana y abuelos, por ser mi mayor fuente de motivación, por impulsarme siempre a superarme y por acompañarme incondicionalmente hasta el final. Los amo con todo mi ser.

Agradecimientos

En primer lugar, expresar mi más sincero agradecimiento a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) por haber proveído los fondos necesarios para la realización de este trabajo de investigación.

Agradezco también a la Universidad Autónoma de Querétaro (UAQ), en especial a la Facultad de Ingeniería, por brindarme el espacio y los conocimientos necesarios a lo largo de mis estudios de posgrado.

Mi más profundo agradecimiento a mi madre Karina Castro Salazar, a mi padre, Alfonso de Jesús Juárez Castellanos y a mi hermana, Paulina Sofía Juárez Castro. Por su constante apoyo y amor, por nunca dejarme solo, y por siempre creer en mí. Sé que todo va a salir bien si existe la fuerza y motivación por parte de ustedes en cada nuevo escalón.

A mi director de tesis, Dr. Marco Antonio Aceves Fernández, por sus enseñanzas, sus valiosos consejos, por confiar en mí y sobre todo por ayudarme a conseguir una mejor versión de mí mismo a lo largo de toda la maestría.

A todo mi sínodo: Dr. Efrén Gorrostieta Hurtado, Dr. Juan Manuel Ramos Arreguín, Dr. Saúl Tovar Arriaga y Dr. Jesús Carlos Pedraza Ortega, por sus enseñanzas, apoyo moral y valiosa orientación en el desarrollo de la investigación. Un agradecimiento especial al Dr. Sebastián Salazar Colores por sus sabios consejos y por compartir su experiencia a lo largo de esta travesía.

A mi querido amigo Imanol Betanzos, por estar siempre a mi lado, brindándome apoyo y consejos en los momentos más necesarios. A Dr. Alan Zabaleta, por su valiosa amistad y por su disposición constante para ayudarme en todo lo que necesitaba. Y, en general, a las grandes amistades que he tenido el privilegio de desarrollar durante mi maestría; les llevaré siempre en mi corazón.

Abreviaturas y siglas

EMG Electromiografía

IA Inteligencia Artificial

CNN Red Neuronal Convolucional

LSTM Memoria a Largo Corto Plazo

EMD Descomposición Modal Empírica

FT Transformada de Fourier

ViT Visual Transformer

MuCBiT Multichannel Biosignal Transformer

Na⁺ Sodio

K⁺ Potasio

Cl⁺ Cloro

mV Milivoltios

MU Unidades Motoras

ML Aprendizaje Máquina

DL Aprendizaje Profundo

NN Red Neuronal

RNN Red Neuronal Redurrente

MLP Perceptron Multicapa

tanh Tangente Hiperbólica

PatchTST Patch Time Series Transformer

Hz Hercios

GHZ Gigahercios

PLI Interferencia de la línea de energía

DFT Transformada de Fourier Discreta

FFT Transformada Rápida de Fourier
WT Transformada Wavelet
CWT Transformada Wavelet Continua
DWT Transformada Wavelet Discreta
MAV Valor Absoluto Medio
MSR Error Cuadrático Medio
VAR Varianza
ReLU Unidad Lineal Rectificada
iDWT Transformada Wavelet Discreta Inversa
GS Búsqueda en cuadrícula
GP Proceso Gaussiano
TP Verdadero Positivo
TN Verdadero Negativo
FN Falso Negativo
FP Falso Positivo
ROC Característica de Operación del Receptor
FPR Tasa de Falsos Positivos
TPR Tasa de Verdaderos Positivos
AUC Área Bajo la Curva
t-SNE Incrustación Estocástica de Vecinos Distribuida en t

Resumen

El cuerpo humano funciona gracias a una red de señales denominadas señales bioeléctricas, las cuales se generan y transmiten continuamente. Estas señales son una herramienta clave para comprender diversos aspectos del funcionamiento biológico y fisiológico del cuerpo humano.

Dentro de las señales producidas por el cuerpo humano se encuentran las señales mioeléctricas, que pueden registrarse mediante electromiografía (EMG). Estas señales proporcionan información sobre la activación y el comportamiento de los músculos. En la actualidad, las señales EMG son un importante objeto de estudio en áreas como la rehabilitación, la medicina del deporte y la neurociencia, entre otras. Sin embargo, su uso no es una tarea fácil, ya que presentan diversos desafíos que deben ser superados para garantizar su correcta interpretación y aplicación.

La inteligencia artificial es una herramienta poderosa capaz de superar estas barreras, permitiendo el desarrollo de modelos que se adapten a la naturaleza dinámica de las señales EMG. En esta investigación se propone analizar las problemáticas presentes en las señales EMG y emplear estrategias para su tratamiento, con el fin de clasificar diez movimientos de la mano utilizando algoritmos de aprendizaje profundo.

Palabras clave: señales bioeléctricas, señales mioeléctricas, EMG, inteligencia artificial, aprendizaje profundo, clasificación de movimientos.

Abstract

The human body functions through a network of signals known as bioelectrical signals, which are continuously generated and transmitted. These signals are a key tool for understanding various aspects of the body's biological and physiological functions. Among the signals produced by the human body are myoelectric signals, which can be recorded using electromyography (EMG). These signals provide information about muscle activation and behavior. Currently, EMG signals are an important subject of study in fields such as rehabilitation, sports medicine, and neuroscience, among others. However, their use is not an easy task, as they present various challenges that must be overcome to ensure their proper interpretation and application.

Artificial intelligence is a powerful tool capable of overcoming these barriers, enabling the development of models that adapt to the dynamic nature of EMG signals. This research aims to analyze the challenges associated with EMG signals and implement strategies for their processing to classify ten hand movements using deep learning algorithms.

Keywords: bioelectric signals, myoelectric signals, EMG, artificial intelligence, deep learning, motion classification.

Índice

Índice de figuras

Índice de cuadros

1	Introducción	1
1.1	Justificación	1
1.2	Descripción del problema	2
1.3	Hipótesis	2
1.4	Objetivo general	2
1.5	Objetivos específicos	3
1.6	Estado del arte	3
2	Fundamentación teórica	5
2.1	Señal bioeléctrica	5
2.2	Señal EMG	8
2.3	Inteligencia Artificial (IA)	9
2.4	Aprendizaje Máquina (ML)	10
2.4.1	Aprendizaje supervisado:	10
2.4.2	Aprendizaje no supervisado:	11
2.5	Aprendizaje Profundo (DL)	11
2.5.1	Redes Neuronales (NN):	11
2.5.2	Redes Neuronales Recurrentes (RNN):	11
2.5.3	Memoria a Largo Corto Plazo (LSTM):	13
2.5.4	Redes Neuronales Convolucionales (CNN):	14
2.5.5	Transformers	15
2.5.6	Patch Time Series Transformer (PatchTST):	17
2.6	Procesamiento de la señal EMG	18
2.7	Ruido en una señal EMG	19
2.8	Reducción de ruido en la señal EMG	19
2.9	Filtros digitales	19
2.9.1	Transformada de Fourier (FT)	21
2.9.2	Transformada Wavelet (WT)	21
2.10	Segmentación de la señal EMG	23
2.10.1	Segmentación por eventos	24
2.10.2	Segmentación por ventanas deslizantes	24
2.11	Extracción de características	25
2.11.1	Extracción de características utilizando CNN	26
2.11.2	Extracción de características utilizando Transformers	27
2.12	Clasificación	28
3	Metodología	30

3.1	Materiales	32
3.2	Base de datos	32
3.3	Procesamiento de la señal	36
3.4	Eliminación de ruido	36
3.5	Segmentación	39
3.6	Clasificación	42
4	Resultados y discusión	46
4.1	Evaluación de métricas	46
4.1.1	Matriz de confusión	46
4.1.2	Accuracy	46
4.1.3	Precisión	47
4.1.4	Recall	47
4.1.5	Especificidad	47
4.1.6	F1-Score	47
4.1.7	Curva ROC y AUC	48
4.1.8	Curvas de pérdida	49
4.1.9	t-SNE	49
4.1.10	P-valor	51
4.2	Resultados	52
5	Conclusiones	63
	Referencias bibliográficas	64
6	Appendices	73

Índice de figuras

2.1	Diferencia de potencial de las células en el reposo. Adaptado de [15]	6
2.2	Intercambio de iones en la fase de despolarización. Adaptado de [15]	6
2.3	Intercambio de iones en la fase de repolarización. Adaptado de [15]	7
2.4	Gráfica del ciclo del potencial de acción. Adaptado de [20]	8
2.5	Comparación de la señal EMG mediante dos técnicas: (a) EMG de superficie, y (b) EMG intramuscular. Adaptado de [26].	9
2.6	Taxonomía de la IA. Adaptado de [30]	10
2.7	Representación gráfica de la red neuronal. Adaptado de [36]	12
2.8	Representación gráfica de la RNN. Adaptado de [36]	12
2.9	Diferencias entre la estructura interna de un bloque en las redes: (a) RNN, y (b) LSTM. Adaptado de [39].	13
2.10	Representación gráfica de la CNN. Adaptado de [43]	14
2.11	Arquitectura del modelo transformer. Extraído de [46]	16

2.12	Arquitecturas PatchTST: (a) Supervisado, y (b) Auto-supervisado. Adaptado de [48].	18
2.13	Cuatro filtros digitales más comunes: (a) Filtro pasa bajas, (b) Filtro pasa altas, (c) Filtro pasa bandas, y (d) Filtro rechaza banda. Adaptado de [62].	20
2.14	Segmentación mediante inicio y final de un evento muscular. Adaptado de [74]	24
2.15	Métodos de segmentación por ventanas: (a) Ventaneo adyacente, y (b) Ventaneo superpuesto. Adaptado de [76, 77].	25
2.16	Extracción de características in una CNN. Adaptado de [83]	26
2.17	Extracción de características en un transformer. Adaptado de [44]	27
3.18	Diagrama de flujo de la metodología.	30
3.19	Lista de movimientos: (a) Posición inicial, (b) Pronación, (c) Supinación, (d) Extensión, (e) Flexión, (f) Desviación cubital, (g) Desviación radial, (h) Pinza fina, (i) Pinza gruesa, (j) Expansión. Adaptado de [85].	34
3.20	Comparación entre la wavelet madre db20 y la señal EMG original.	36
3.21	Descomposición en niveles mediante DWT. Adaptado de [89]	37
3.22	Comparación entre las señales del movimiento de flexión: (a) Señal bruta, y (b) Señal procesada.	38
3.23	Segmentación de los eventos basados en el tiempo de recolección.	39
3.24	Agrupación de las muestras tras aplicar el ventaneo en cada segmento.	41
4.25	Representación de una matriz de confusión. [97]	46
4.26	Representación de una curva ROC. Adaptado de [100]	48
4.27	Representación de una curva de pérdida. Adaptado de [101]	49
4.28	Proceso de reducción de la dimensionalidad mediante t-SNE: (a) Medición de las distancias entre el punto de interés y los demás puntos, (b) Muestreo de las distancias en una distribución gaussiana, (c) Cálculo de la puntuación de similitud entre los puntos. Adaptado de [102].	50
4.29	Curvas de rendimiento del clasificador CNN.	52
4.30	Curvas de rendimiento del clasificador CNN-LSTM.	53
4.31	Curvas de rendimiento del clasificador PatchTST.	53
4.32	Matriz de confusión del modelo CNN	54
4.33	Matriz de confusión del modelo CNN-LSTM	54
4.34	Matriz de confusión del modelo PatchTST	54
4.35	Gráfico de barras de las métricas de rendimiento del modelo CNN.	57
4.36	Gráfico de barras de las métricas de rendimiento del modelo CNN-LSTM.	57
4.37	Gráfico de barras de las métricas de rendimiento del modelo PatchTST.	58
4.38	Curva ROC del modelo CNN.	58
4.39	Curva ROC del modelo CNN-LSTM.	59
4.40	Curva ROC del modelo PatchTST.	59
4.41	t-SNE para el modelo CNN.	61
4.42	t-SNE para el modelo CNN-LSTM.	61

Índice de cuadros

1.1	Trabajos relacionados	4
2.2	Lista de la 324 wavelets madre ordenadas en 15 familias. Adaptado de [3]	23
3.3	Descripción de cada músculo implicado en los movimientos. [84]	31
3.4	Distribución de los sensores de la Myo armband en cada músculo. Adaptado de [85]	33
3.5	Sensores involucrados en los movimientos. Adaptado de [87]	35
3.6	Tiempos considerados para cada repetición del evento	39
3.7	Total de señales con datos no segmentados y segmentados.	40
3.8	Total de señales con datos no segmentados, segmentados y con ventaneo.	41
3.9	Primeros hiperparámetros utilizados	44
3.10	Comparación entre etiquetas	45
4.11	Comparación del accuracy obtenido para cada modelo.	55
4.12	Métricas de rendimiento para el modelo CNN.	56
4.13	Métricas de rendimiento para el modelo CNN-LSTM.	56
4.14	Métricas de rendimiento para el modelo PatchTST.	56
4.15	Valores AUC para el clasificador CNN.	60
4.16	Valores AUC para el clasificador CNN-LSTM.	60
4.17	Valores AUC para el clasificador PatchTST.	60
4.18	Comparación de los accuracy obtenidos en el estado del arte y este trabajo.	61
4.19	Comparación del P-valor obtenido para los tres modelos.	62

1. Introducción

Las señales EMG son parte fundamental dentro de los avances tecnológicos. Estas señales, que provienen de la actividad eléctrica generada por los músculos cuando se realiza un movimiento o una acción, han demostrado ser una herramienta invaluable para una variedad de aplicaciones. Al ser una tecnología que aprovecha directamente la información del cuerpo humano, se ha integrado en diversos campos de la medicina, la rehabilitación y la ingeniería. El procesamiento de las señales mioeléctricas ha dado paso al control de dispositivos, ofreciendo un movimiento más natural, pues la señal proviene directamente del usuario. [1]

Apesar de presentar grandes avances tecnológicos y un estado del arte amplio, el uso y aplicación de las señales EMG sigue presentando un gran desafío. Esto debido a la naturaleza de la señal, siendo esta no estacionaria, ruidosa y de varianza entre sujetos por características como el tamaño y forma de los músculos, la dieta del sujeto, e incluso la cantidad de grasa presente. [2, 3] Además, de la presencia de factores externos que dificultan el uso de las señales EMG como lo son el ruido ambiental, los artefactos de movimiento al momento de la obtención de la señal, entre otros.

El uso de consideraciones basadas en herramientas actuales para la extracción de las señales, constituye una mejora significativa pues facilita la interacción directa del sensor con el músculo, así como la atención a distintos grupos musculares partícipes en el movimiento.

Dentro de los sistemas EMG para el reconocimiento de movimientos se considera el procesamiento de los datos, la segmentación, la extracción de características y el uso de las mismas para su clasificación. [4] Gracias a la presencia de la inteligencia artificial, estos sistemas pueden resumirse en un enfoque más eficiente y preciso, donde los algoritmos de aprendizaje automático y redes neuronales son capaces de identificar patrones complejos en los datos, mejorar la clasificación de movimientos y adaptarse a la varianza de los sujetos. Esto permite un rendimiento optimizado en aplicaciones como el control de prótesis, la rehabilitación y la interacción humano-máquina.

Siguiendo el enfoque, este trabajo se centra en la implementación de técnicas de procesamiento de señales, buscando extraer las características relevantes que alimenten algoritmos de aprendizaje profundo para mejorar la clasificación de diez movimientos de la mano utilizando señales obtenidas del brazo derecho de cincuenta usuarios. Considerando características como la diversidad de los usuarios en cuanto a edad y género.

1.1. Justificación

La amputación de un miembro irreparablemente dañado es una de las operaciones quirúrgicas más antiguas y graves [5], presentes a día de hoy y afectando a más de veinte mil personas al año en México.[6] Las señales EMG generadas por la actividad eléctrica de los músculos al contraerse, ofrecen una fuente valiosa de información que puede ser aprovechada para mejorar significativamente la calidad de vida de las personas con discapacidades físicas, como amputaciones o lesiones medulares.

Una de las tantas problemáticas existentes al trabajar con señales EMG, es la varianza entre sujetos. De acuerdo al estado del arte, existen avances bastante significativos con respecto al uso de las señales. Es posible entonces, utilizar técnicas actuales que permiten la optimización en tiempo y forma de trabajo para controlar las señales EMG, produciéndose metodologías que sirvan como enfoque al desarrollo de alternativas viables para el control de prótesis o rehabilitación en pacientes. Esto promueve el uso de algoritmos de inteligencia artificial para resolver el conjunto de problemas presentes al trabajar con señales EMG y en última instancia desarrollar alternativas que impactan en la calidad de vida del paciente.

1.2. Descripción del problema

El control mioeléctrico es una de las aplicaciones más populares y relevantes en el campo de la biomecánica y la ingeniería biomédica. Utilizando las señales EMG como factor principal para operar dispositivos externos. Para ello, es necesario obtener un conjunto de señales que sean aptas para su clasificación, lo cual implica un proceso complejo que abarca varias etapas clave. [7] Algoritmos que sean capaces de realizar una clasificación exitosa basada en las señales depende de la metodología aplicada, combinando técnicas de procesamiento de señales, extracción de características, segmentación de datos, entre otros. [8] Como bien hemos mencionado, utilizar las señales EMG para llegar a un control adecuado es una tarea compleja afectada directamente por la naturaleza de la señal. Es necesario considerar alternativas más efectivas, que optimicen y reduzcan el tiempo de desarrollo. Es por ello que, se propone el uso de metodologías basadas en algoritmos de aprendizaje profundo para mezclar las técnicas de procesamiento de señales con algoritmos capaces de extraer características y realizar una clasificación basada en las mismas.

1.3. Hipótesis

La implementación de algoritmos de aprendizaje profundo representa una oportunidad significativa para mejorar la precisión y eficiencia para la clasificación de las señales en movimientos. Bajo este enfoque, la mezcla de metodologías de procesamiento con algoritmos de aprendizaje profundo tienen el potencial de superar los desafíos convencionales, como el ruido, la variabilidad entre individuos y la complejidad dimensional.

1.4. Objetivo general

Realizar el procesamiento de las señales para generar la clasificación de diez movimientos de la mano provenientes de la extremidad superior derecha basadas en señales EMG utilizando algoritmos de aprendizaje profundo.

1.5. Objetivos específicos

- I. Verificar las señales mioeléctricas con respecto a los movimientos para comprender la naturaleza de los datos que permitirán clasificar el movimiento.
- II. Dividir las señales en porciones temporales que representen diferentes etapas o características de la actividad registrada como preparación para el análisis.
- III. Procesar los datos para limpiar de ruidos o alteraciones presentes en la señal.
- IV. Obtener las características de la señal con respecto al movimiento.
- V. Implementar algoritmos de aprendizaje profundo teniendo como entrada la señal EMG procesada para clasificarlas con respecto al movimiento.

1.6. Estado del arte

El uso de las señales EMG como solución para abordar problemáticas cotidianas ha experimentado un notable avance en las últimas décadas, gracias al desarrollo de tecnologías emergentes y la implementación de metodologías basadas en algoritmos complejos. Estas señales han encontrado aplicaciones en una amplia gama de áreas, incluyendo la rehabilitación, la biomecánica, el control de prótesis, entre otras. Consolidándose como una herramienta esencial en el ámbito de la investigación y la innovación tecnológica. Se han llevado a cabo trabajos que buscan mejorar la calidad de vida de los usuarios. Sin embargo, aún existe un amplio margen para optimizar estas tecnologías. A continuación se describen algunos trabajos que impulsan el desarrollo de tecnologías capaces de mejorar los retos presentes al trabajar con señales EMG.

Cuadro 1.1: Trabajos relacionados

Ref.	Contribución	Metodología utilizada	Resultados
Too, J. et al. (2019) [9]	Reconocimiento de patrones sin extracción manual de características para 17 movimientos de mano y muñeca.	Utilizando la base de datos Nina-pro, las señales se convirtieron en espectrogramas para alimentar dos modelos basados en la arquitectura de Redes Neuronales Convolucionales (CNN), los cuales realizaron la clasificación de 17 movimientos de mano y muñeca.	Accuracy 1: 86.18 % Accuracy 2: 88.04 %
Huang, D. & Chen, B. (2020) [10]	Reconocimiento de gestos de la mano mediante la clasificación de 49 gestos.	Utilizando el conjunto de datos Nina-pro, se utilizaron los datos correspondientes a 49 gestos distintos para clasificarlos mediante una combinación de una red CNN para la extracción de características y una red de Memoria a Largo Corto Plazo (LSTM) para mantener la información secuencial de los datos.	Accuracy: 79.329 %
Bakircioğlu, K. & Ozkurt, N. (2020) [11]	Clasificación de 6 movimientos de la mano utilizando una CNN y Descomposición Modal Empírica (EMD).	La clasificación de datos de 6 movimientos de la mano de 5 sujetos, obtenidos del UCI Machine Learning Repository, se realizó aplicando diversas técnicas de procesamiento, como una ventana deslizante para enfocarse en el segmento donde ocurre la contracción, una Transformada de Fourier (FT) para reducir el ruido y la extracción de características mediante una CNN.	Accuracy: 95.90 %
Núñez, B. et al. (2023) [12]	Clasificación de cuatro gestos de la mano para controlar una mano robótica, utilizando un modelo basado en Visual Transformer (ViT) llamado Multichannel Bio-Signal Transformer (MuCBiT).	Basándose en la metodología de un ViT, los autores presentan el modelo MuCBiT con el objetivo de realizar el control de una mano robótica utilizando un controlador de servomotores para ejecutar cuatro gestos. El conjunto de datos (adquirido por los propios autores) es completamente público. Los datos fueron obtenidos de diez participantes utilizando un dispositivo Cyton Board.	Accuracy: 86.78 %

2. Fundamentación teórica

2.1. Señal bioeléctrica

Al igual que las baterías, el cuerpo humano genera y utiliza diferencias de carga eléctrica para realizar diversas funciones. En una batería, las cargas eléctricas se separan entre los polos, creando un potencial eléctrico. De manera similar, en nuestras células, la membrana celular actúa como una barrera semipermeable que separa dos entornos con diferentes concentraciones de iones, formando un gradiente electroquímico. [13, 14]

La membrana celular es una estructura que rodea y delimita todas las células vivas. Dentro, la membrana separa diferentes concentraciones de iones, entre los que destacan el Sodio (Na^+), el Potasio (K^+) y el Cloro (Cl^-). Estos, junto a determinadas proteínas como la bomba Na^+/K^+ dan lugar al denominado gradiente de potencial. [15] El potencial de acción se genera cuando la célula es estimulada. Este proceso se repite de manera secuencial a lo largo de la membrana celular. Esta propagación secuencial del potencial de acción permite la transmisión de señales eléctricas a lo largo de las células excitables neuronas y fibras musculares, facilitando la comunicación entre diferentes partes del cuerpo. [16] La repetición de este proceso se produce de forma rápida y coordinada, permitiendo una transmisión eficiente de las señales eléctricas a través de las células excitables, dando como resultado una señal bioeléctrica.

Considerando la actividad eléctrica del cuerpo como la diferencia de voltaje existente a través de la membrana celular, esta se divide en tres fases: el potencial en reposo, la despolarización y la repolarización. [13, 17, 18]

- a) **Potencial en reposo:** Cuando la neurona se encuentra en su 'posición inicial', es decir, cuando la carga eléctrica es más negativa en el interior de la membrana y más positiva en el exterior, se presenta el potencial en reposo. Este estado se logra debido a la distribución desigual de iones a través de la membrana celular. En el interior de la célula, existen iones negativos (aniones), como proteínas cargadas negativamente y fosfatos, que no pueden salir de la célula debido a su tamaño o a su naturaleza. En menor concentración, se encuentran iones de Cl^- , que también ayudan a mantener el equilibrio de cargas.

Por lo tanto, la diferencia de voltaje existente en la membrana no genera impulsos eléctricos (o potenciales de acción), pero mantiene la célula en un estado preparado para generarlos cuando sea necesario. Esto se debe a la distribución desigual de iones como Na^+ y K^+ , junto con los aniones presentes en el interior.

El potencial en reposo se sitúa entre -60 mV y -100 mV, dependiendo del tipo de célula.

La Figura 2.1 describe la distribución de cargas en el exterior e interior de la célula. La membrana celular cuenta con varios canales iónicos y la bomba de Na^+/K^+ . Sin embargo, los dos canales más relevantes son los canales de K^+ , los cuales permiten,

como su nombre lo indica, que los iones de K^+ salgan de la célula, mateniéndose así el potencial negativo en el interior de la célula durante el estado de reposo, y los canales de Na^+ , que permiten la entrada de iones de Na^+ , especialmente cuando la célula se despolariza, es decir, cuando se presenta un cambio en el potencial eléctrico de la membrana celular, por lo que, durante el reposo, se encuentran cerrados.

Ambos son de vital importancia para la estabilidad de la polaridad de la membrana.

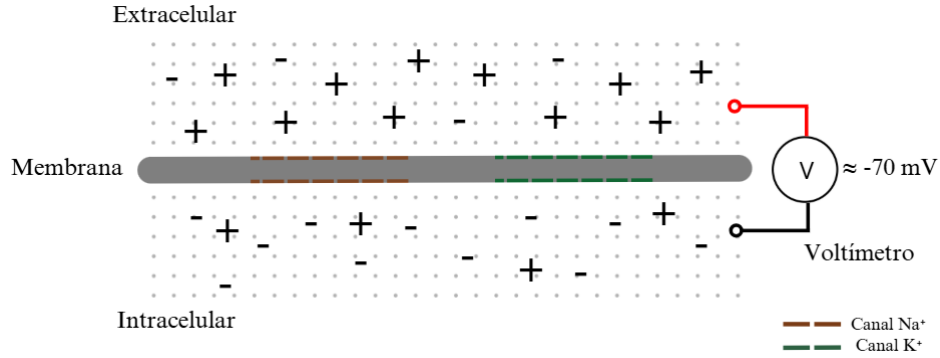


Figura 2.1: Diferencia de potencial de las células en el reposo. Adaptado de [15]

- b) **Despolarización:** El proceso de despolarización se produce gracias a un cambio en la permeabilidad de la membrana celular ocasionado por la activación de canales iónicos. Todo parte en el momento que se genera un estímulo, esto produce un cambio en el voltaje de la membrana. Si dicho cambio en el voltaje producido por el estímulo es lo suficientemente fuerte para alcanzar el umbral de excitación ($\approx -55 \text{ mV}$), se activan los canales de Na^+ (los cuales permiten la entrada masiva de iones positivos a la célula). Este flujo de iones al interior de la célula reduce la carga negativa en el interior, así como también reduce la carga positiva en el exterior de la célula, desencadenándose la despolarización.

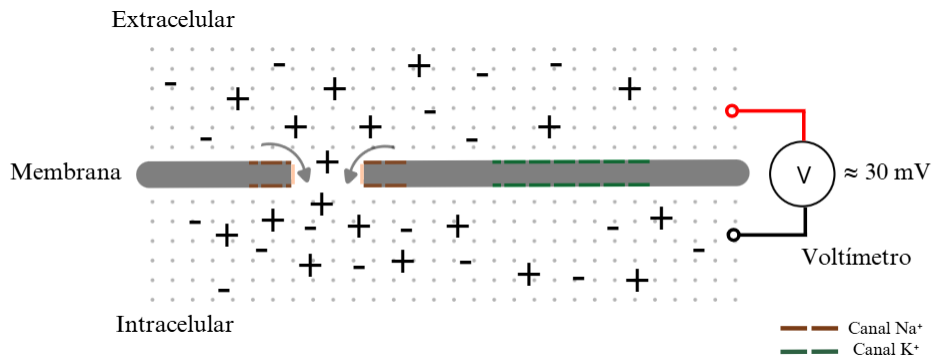


Figura 2.2: Intercambio de iones en la fase de despolarización. Adaptado de [15]

La Figura 2.2 describe gráficamente el intercambio de cargas presentes en la fase de despolarización.

Cabe aclarar que no existe un movimiento activo de cargas negativas en la fase de despolarización. Sin embargo, al ingresar Na^+ (el cual contiene carga positiva), al interior de la célula, esta pierde la predominancia de carga negativa, alcanzándose cargas con valor positivo. Es importante considerar que los canales de Na^+ están formados por dos puertas. La puerta de activación y la puerta de inactivación. La primera se abre al alcanzar el umbral de activación y permite la entrada de la carga positiva. La segunda puerta, regula o detiene el flujo de estas cargas después de la activación, generalmente, cuando se alcanza el pico del potencial de acción, que es de aproximadamente 30mV .

- c) **Repolarización:** En esta fase, se presenta la salida de la carga positiva (que ingresa en la fase de despolarización) desde el medio intracelular hacia el medio extracelular, restableciendo el potencial de membrana hacia valores negativos. Los iones de K^+ (que ya se encontraban en alta concentración en el medio intracelular como producto de la bomba Na^+/K^+), ahora se dirigen al medio extracelular a través de los canales de K^+ . Este flujo de iones positivos hacia el exterior compensa o regula el cambio en la distribución de cargas causada por la entrada masiva de iones de Na^+ durante la despolarización. Como resultado, la célula recupera su polaridad inicial, es decir, regresa al potencial en reposo.

En la Figura 2.3, se describe gráficamente la regulación de iones en la fase de repolarización.

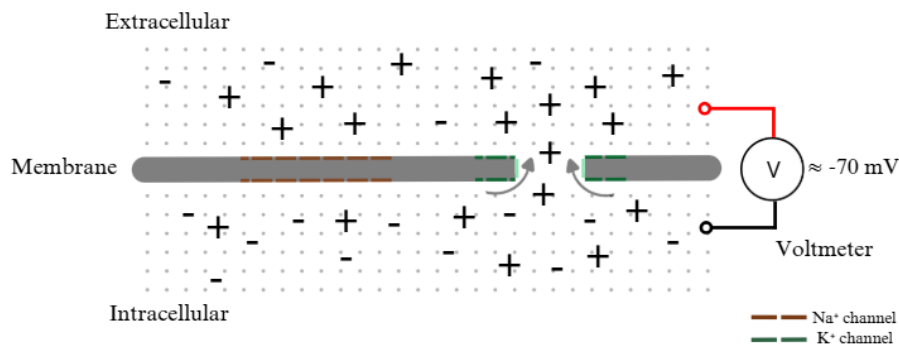


Figura 2.3: Intercambio de iones en la fase de repolarización. Adaptado de [15]

Dentro de la fase de repolarización puede presentarse posterior a ella una fase llamada hiperpolarización. En la hiperpolarización el potencial de la membrana se vuelve más negativo que el valor presente en reposo. Es decir, en la hiperpolarización puede llegarse a valores aproximados de -90mV . Esto debido a que el canal de K^+ puede permanecer abierto más tiempo del necesario, lo que provoca que el interior de la célula se vuelva más negativo.

La repetición del proceso que constituye el potencial en reposo, la despolarización y la repolarización da lugar al fenómeno potencial de acción, este ocurre de manera rápida y coordinada en las células excitables, como las neuronas y las fibras musculares. [19] Permittiéndose la transmisión eficiente de señales eléctricas a través de las membranas celulares, dando lugar a la señal bioeléctrica, la Figura 2.4 muestra cómo se representa la gráfica del ciclo de potencial de acción.

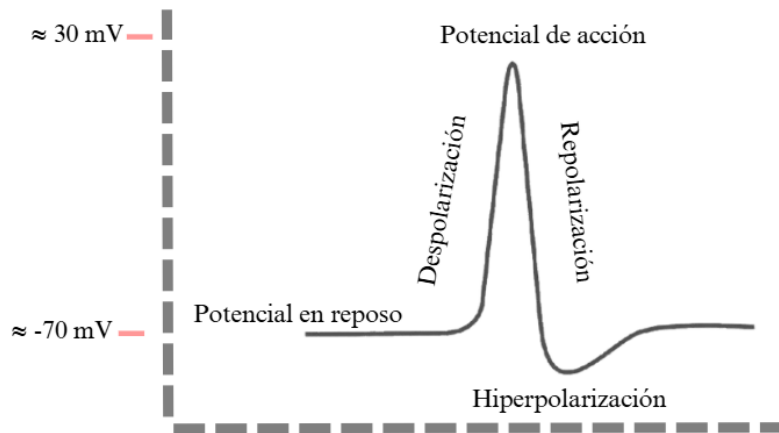


Figura 2.4: Gráfica del ciclo del potencial de acción. Adaptado de [20]

La generación de señales bioeléctricas en los distintos órganos del cuerpo humano sigue un proceso fundamentalmente similar. Aun así, debido a las características específicas de cada sistema, se denominan de forma diferente según el órgano de interés. En el contexto de los músculos, el estímulo que genera el potencial de acción da lugar a una contracción muscular. Es decir, la contracción ocurre gracias a miles de fibras musculares generando potenciales de acción de manera repetitiva y coordinada. Estas señales eléctricas individuales se suman, dando lugar a lo que conocemos como señal EMG.

2.2. Señal EMG

Las señales EMG son la actividad eléctrica generada por los músculos estriados, quienes son los encargados de realizar movimientos corporales como caminar, mover los brazos o incluso expresar emociones faciales. [21, 22] La actividad eléctrica generada por la contracción muscular es la encargada de generar el estímulo para el potencial de acción. Las señales mioeléctricas permiten comprender los movimientos intencionales e incluso aquellos que son automáticos. [23] La técnica utilizada para extraer las señales directamente del músculo que ejerce el movimiento a analizar se conoce como Electromiografía, dicha técnica consiste en la colocación de electrodos o en la inserción de una aguja en el o los músculos para obtener la señal eléctrica. [24] Dentro de los músculos estriados, se encuentran las unidades motoras (MU). Las MU son los grupos funcionales de células que son las principales encargadas de generar la contracción. Las

MU se componen de una motoneurona y de las fibras musculares que la motoneurona se encarga de controlar. [25] El uso de electrodos de superficie o electrodos intramusculares depende de los objetivos. Sin embargo, se ha demostrado que no se encuentra una diferencia significativa al utilizar los distintos métodos en las aplicaciones actuales de las señales EMG, como es el caso de la clasificación de movimientos o de fenómenos donde se implica el movimiento de un músculo. En la Figura 2.5, se presenta la comparación de una señal extraída por: (a) electrodos de superficie, y (b) por electrodos intramusculares.



Figura 2.5: Comparación de la señal EMG mediante dos técnicas: (a) EMG de superficie, y (b) EMG intramuscular. Adaptado de [26].

En la práctica, el uso de múltiples canales o sensores en el análisis de señales EMG es fundamental para obtener una representación más completa y precisa de la actividad muscular. Cada canal de registro proporciona información de una región específica del músculo o grupo muscular, y trabajar con varios canales permite captar mejor la complejidad de las señales. [27]

2.3. Inteligencia Artificial (IA)

La IA nace de la unión de dos palabras: artificial e inteligencia, donde artificial significa no real o no natural e inteligencia se interpreta como la capacidad de razonar, de percibir y aprender. Bajo este contexto, la IA se define como el campo interdisciplinario que abarca diversas áreas como la informática, las matemáticas, la ingeniería, la lingüística, la neurociencia, entre otras. con el objetivo de crear máquinas inteligentes que asemejan al comportamiento humano. [28]

En el año 1955, John McCarthy inventó el término IA bajo la lógica de si una computadora puede realizar una tarea, entonces una calculadora automática puede ser programada para simular a la máquina, atribuyéndose, como el fundador de la IA. [29] En las últimas décadas se han desarrollado avances significativos con respecto a la IA lo que lleva a la creación de algoritmos o metodologías capaces de resolver problemáticas de gran complejidad.

Es importante mencionar las categorías en las que la IA puede ser dividida, en la Figura 2.6 se representa la división de categorías con respecto a los enfoques y técnicas de aplicación.



Figura 2.6: Taxonomía de la IA. Adaptado de [30]

2.4. Aprendizaje Máquina (ML)

La metodología de ML tiene como objetivo permitir que las computadoras aprendan a partir de datos, sin necesidad de ser programadas explícitamente para realizar tareas específicas. En lugar de seguir instrucciones programadas directamente, las computadoras ajustan modelos a partir de patrones y regularidades en los datos, mejorando su desempeño a medida que obtienen más información. [31]

En consecuencia, el problema central radica en la necesidad de desarrollar soluciones más precisas, personalizadas y seguras para la clasificación y el control de señales EMG, superando las limitaciones actuales y promoviendo la colaboración interdisciplinaria para maximizar su potencial en beneficio de las personas con discapacidades y de la sociedad en su conjunto.

ML puede ser clasificado en función de si la máquina maneja un aprendizaje supervisado o un aprendizaje no supervisado.

2.4.1. Aprendizaje supervisado:

El aprendizaje supervisado es aquel tipo de aprendizaje donde se conoce el valor del resultado, es decir, con entradas y sus respectivas salidas conocidas. Entre las técnicas más comunes se incluyen los árboles de decisión, que dividen los datos en subgrupos. Las máquinas de soporte vectorial, que buscan separar las clases mediante un hiperplano. La regresión lineal, que predice un valor continuo en función de variables independientes. Entre otros.

Cada una de estas técnicas es adecuada según el tipo de problema, como clasificación o regresión, y la naturaleza de los datos. [32]

2.4.2. Aprendizaje no supervisado:

En el algoritmo no supervisado se intentan encontrar relaciones o agrupaciones entre los datos sin tener alguna referencia o respuesta. En este caso, se busca encontrar patrones o similitudes entre los datos de entrada, el enfoque se basa en agrupaciones o clustering y reducción de la dimensionalidad de los datos. Algunos ejemplos son las agrupaciones k-means y las agrupaciones de maximización de expectativas. [33]

2.5. Aprendizaje Profundo (DL)

Tal como lo describe la Figura 2.6, el DL es un subconjunto del ML. Se trata de una 'cascada' de múltiples capas de procesamiento que se encargan de extraer y transformar las características para alimentar a capas sucesoras, la división de la cascada se da entre capas de entrada, capas ocultas y capas de salida. Las capas cercanas a la entrada se encargan de aprender características simples, las capas más profundas aprenden características derivadas de las capas inferiores, las capas profundas se encargan de las características más complejas. [34] El DL es especialmente aplicado en aquellas tareas donde se tiene una gran cantidad de datos, lo que vuelve ideal la metodología jerárquica. [35] Desde el inicio se han presentado varios modelos de DL, a continuación se presentan algunos modelos que permiten la clasificación de señales.

2.5.1. Redes Neuronales (NN):

Las NN se inspiran en la sofisticada funcionalidad de los cientos de miles de neuronas de los cerebros humanos. Una NN consta de una capa de entrada de neuronas también conocidas como nodos, consta también de capas ocultas y capas de salida. Cada nodo o neurona se conecta una a otra, dicha unión recibe el nombre de peso, la estructura básica de una NN es descrita en la Figura 2.7. Las NN deben ser entrenadas para que el objetivo se cumpla lo más eficiente posible. Para llevar a cabo esa tarea la red debe recibir datos de entrada, o también conocidos como conjunto de datos de entrenamiento. Los datos de salida deseados deben ser conocidos, de esta forma, el objetivo de la etapa de entrenamiento es minimizar el error de los datos, esto se realiza ajustando los pesos de las conexiones entre neuronas. [36]

2.5.2. Redes Neuronales Recurrentes (RNN):

Las RNN son una arquitectura de red neuronal utilizada principalmente para detectar patrones en secuencias de datos. Estas redes son especialmente útiles cuando la entrada consiste en secuencias, como textos, series de tiempo numéricas, o señales que cambian con respecto al tiempo. [37]

Las RNN a menudo se comparan con las redes neuronales feedforward como los Perceptrones Multicapa (MLP). La principal diferencia radica en cómo se transmite la información. Mientras que las redes MLP transmiten la información en una dirección, sin ciclos, las RNN poseen conexiones recurrentes que permiten que la información se

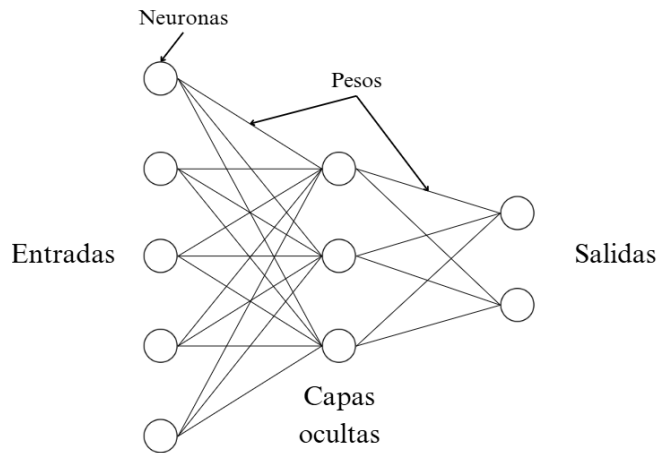


Figura 2.7: Representación gráfica de la red neuronal. Adaptado de [36]

recircule”, es decir, la salida de una capa puede volver a ser utilizada como entrada en pasos posteriores. Esta retroalimentación permite que las RNN mantengan información sobre entradas previas, lo que les otorga la capacidad de manejar secuencias de datos a lo largo del tiempo, a diferencia de las redes MLP, que solo procesan la entrada actual. [38]

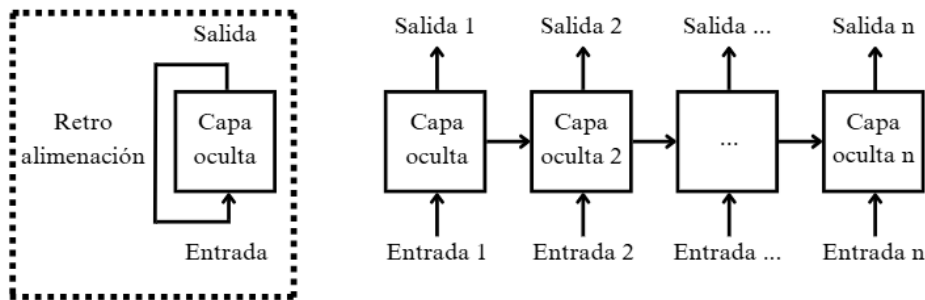


Figura 2.8: Representación gráfica de la RNN. Adaptado de [36]

En la Figura 2.8 se detalla la arquitectura de la RNN, la primera capa oculta recibe una entrada, ya sea un vector de características o un vector escalar y esta información es procesada dentro de la capa, generando una representación intermedia. Como no presenta datos de tiempo anteriores para generar una retroalimentación, se retroalimenta a sí misma en el mismo paso de tiempo. Como siguiente paso, la capa oculta genera una salida, misma que sirve como entrada para la siguiente capa. Este proceso se repite en cada paso del tiempo, permitiendo así considerar toda la secuencia de datos. [38] Las RNN tienen gran relevancia en aplicaciones como el tratamiento de datos secuenciales y el tratamiento de estructuras de datos como textos, videos y audios. [39]

2.5.3. Memoria a Largo Corto Plazo (LSTM):

La RNN sirve como arquitectura base para nuevos modelos, pues dentro de su uso, se presentan problemáticas como el aprendizaje de dependencias a largo plazo. [40, 41] Las LSTM se presentan como una alternativa a esta problemática, introduciendo un elemento llamado célula de memoria. Siendo un pequeño sistema autónomo que puede olvidar, recordar y procesar información de forma más compleja. En la Figura 2.9 se presenta la comparación entre la estructura interna de un bloque entre una RNN y una LSTM.

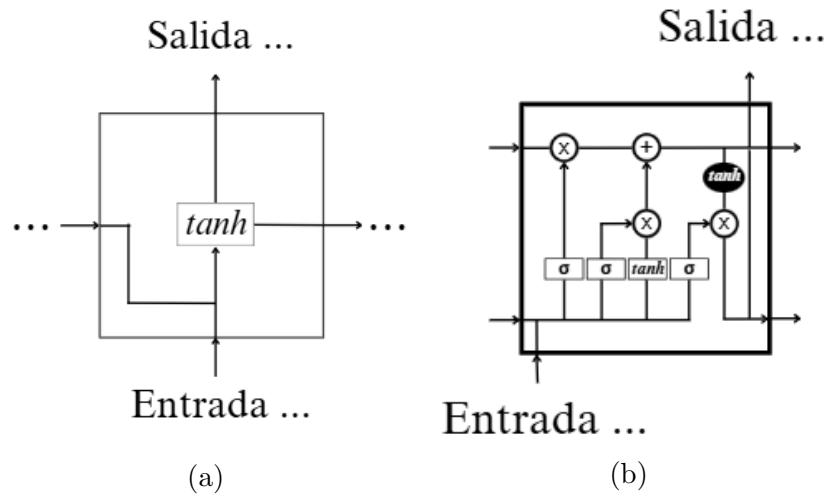


Figura 2.9: Diferencias entre la estructura interna de un bloque en las redes: (a) RNN, y (b) LSTM. Adaptado de [39].

Tal como lo describe la Figura 2.9, la RNN internamente cuenta únicamente con la función de activación tangente hiperbólica (\tanh), esto para tener un control de la información que se procesa entre bloques, transformando el estado de memoria. En la red LSTM, se utilizan dos componentes principales; tres puertas (basadas en funciones sigmoide) que controlan el flujo de información y la transformación del estado de memoria. La primera puerta (primera función sigmoide), llamada puerta de olvido, regula cuánto del estado de memoria previo debe ser olvidado. La segunda puerta (segunda función sigmoide), llamada puerta de entrada, controla cuánto de la nueva información candidata debe añadirse al estado de memoria. La salida de la puerta de entrada se multiplica por la información candidata generada por la función \tanh . Esta multiplicación regula cuánta de la nueva información se almacenará en el estado de memoria, es con esta operación que se calcula la cantidad de información que se almacena en la memoria, considerando también la salida de la puerta de olvido, para ajustar con la información previamente olvidada. La última puerta (tercera función sigmoide), la cual decide qué cantidad de información se utilizará para generar el siguiente estado oculto. Es decir, la información que será enviada al siguiente bloque de trabajo, considerando el ajuste mediante operaciones de multiplicación y la función \tanh .

Finalmente, la información se envía de dos formas principales, considerando la información relevante procesada, permitiendo que el modelo tenga en cuenta el contexto de los datos hasta ese momento. Y la segunda salida permite recordar la información importante de los pasos anteriores. Siendo entonces una salida para la información procesada y una salida para la memoria de dicha información (contexto a largo plazo). [42, 43]

2.5.4. Redes Neuronales Convolucionales (CNN):

La CNN recibe su nombre gracias a una operación matemática entre matrices llamada convolución. Las CNN constan de varias capas: capa de convolución, capa de pooling y la capa completamente conectada [40]. La capa de convolución es utilizada para la extracción de características, cada entrada de la neurona en la capa de convolución está conectada a un campo receptivo local de la neurona anterior. Esto permite que la CNN capture patrones locales en los datos de entrada. La capa de pooling se encarga de realizar un mapeo de características aplicando filtros o kernels para detectar aquellas características específicas en los datos previamente procesados. El objetivo de la capa de pooling es la de reducir la dimensionalidad de los datos conservando la información importante. En la capa completamente conectada se sigue la misma metodología que una red neuronal tradicional por lo que cada nodo está conectado directamente a cada nodo de la capa anterior y a la siguiente. [37, 39, 43]

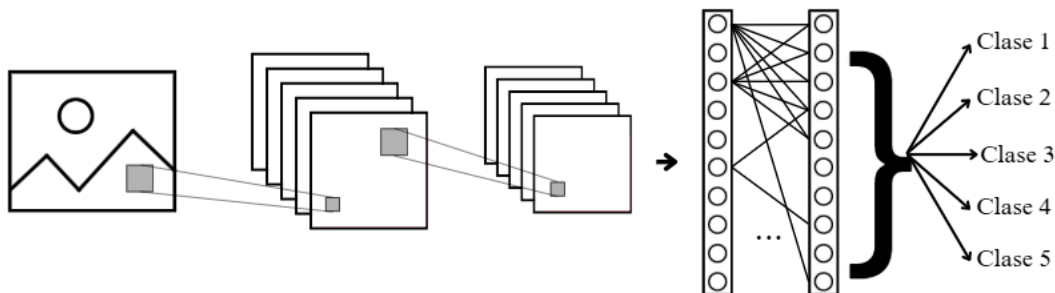


Figura 2.10: Representación gráfica de la CNN. Adaptado de [43]

En la Figura 2.10 se presenta la estructura básica de una CNN. Las primeras capas (capas de convolución) extraen características de la imagen para posteriormente tener su salida como entrada hacia el segundo conjunto de capas (capas de pooling), capas que se encargará de reducir la dimensionalidad de los datos en su entrada. Para posteriormente ingresar a las capas completamente conectadas que se encargarán de establecer la relación entre datos, encontrando patrones de coincidencia y de esta forma teniendo una salida de clasificación. En la salida puede apreciarse que se tienen diversos tipos de clasificaciones dependiendo del tipo de imagen que se ingrese, en el ejemplo, la salida debe clasificarse en el grupo de acuerdo a las características encontradas en la imagen. [43]

2.5.5. Transformers

Apesar de haberse presentado una solución a las problemáticas en los datos secuenciales, se continuaban presentando limitaciones con respecto a la captura de las dependencias a largo plazo y sobre todo en entrenamientos lentos producto de la naturaleza de los datos. Es entonces que, en [44] se presentó la metodología Transformers, una alternativa para solucionar las problemáticas existentes abordando muchas de estas limitaciones.

Los Transformers no solo eliminaron la dependencia de procesar secuencias de manera estrictamente ordenada, sino que introdujeron un nuevo paradigma basado en el mecanismo de atención, una técnica utilizada para dar pesos a distintas partes de una secuencia de entrada de modo que se logre una mejor comprensión de su contexto. [45]

De acuerdo a la metodología, el mecanismo de atención opera con tres valores: *Queries* (Q), *Keys* (K) y *Values* (V). Q , es el dato de interés, es decir, la información con la que queremos buscar relevancia dentro de la metodología del transformer. K , es el valor con el que se compara Q para encontrar relevancia. Finalmente, V es la información que se pondera según la semejanza entre Q y K , proporcionando el resultado de la atención.

Para la conversión del valor de entrada en las tres representaciones distintas se realiza una multiplicación de la matriz inicial con pesos de entrenamiento, los cuales, como hemos visto, se inician de forma aleatoria. Para ello, se utiliza la ecuación (1).

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = X \begin{bmatrix} W_Q \\ W_K \\ W_V \end{bmatrix} \quad (1)$$

Donde X , representa cada matriz de entrada el cual se multiplica por la matriz de peso correspondiente.

La similitud entre Q y K se calcula mediante la ecuación (2).

$$\text{Atención}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) \quad (2)$$

La ecuación muestra una división entre el producto punto de Q y K , escalado por $\sqrt{d_k}$, la cual nos permite evitar valores muy grandes en el cálculo de la similitud. Pues, también se aplica una conversión de valores a probabilidades usando la función softmax (lo cual escala el valor entre 0 y 1). Para finalmente, utilizar la ecuación (3), la cual nos permite tener la representación final del mecanismo de atención.

$$\text{Atención} \times V \quad (3)$$

Estos tres valores son la fase principal de las cabezas de atención, ya que es en base a ellos que el modelo captura diferentes tipos de relaciones o dependencias en la entrada. La cantidad de cabezas de atención a asignar depende del diseño del modelo, y su elección influye en la capacidad del modelo para aprender diversas representaciones de la entrada.

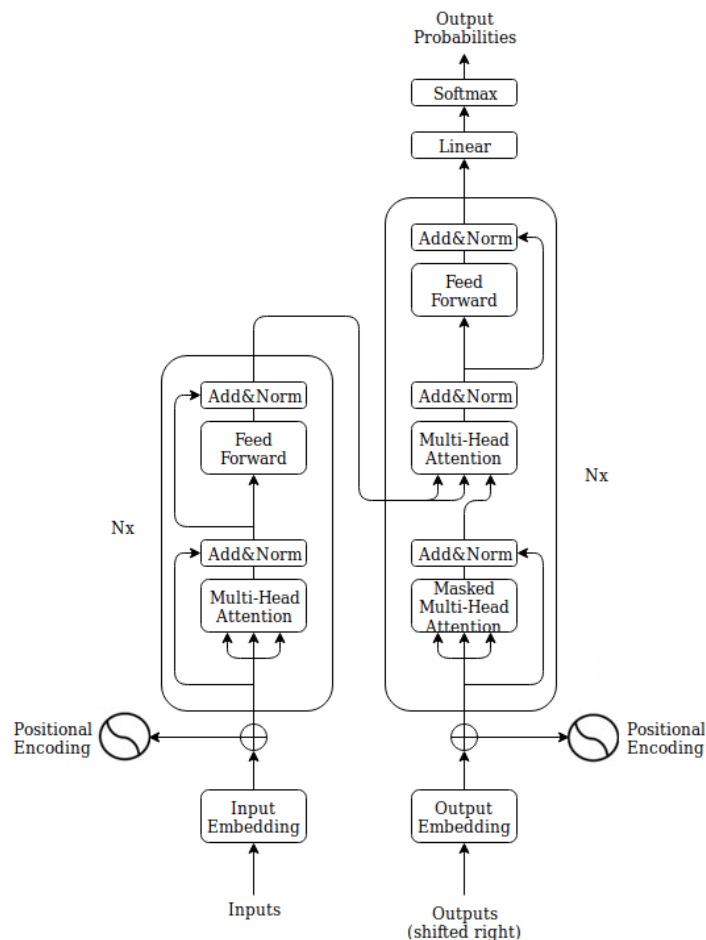


Figura 2.11: Arquitectura del modelo transformer. Extraído de [46]

La Figura 2.11 representa la arquitectura general del modelo Transformer. Esta se divide en dos bloques principales: el codificador y el decodificador. El codificador (primer bloque) recibe una entrada, dicha entrada serán los datos de interés. Los datos son convertidos en vectores numéricos llamados embeddings. Posteriormente a la conversión de los datos en embeddings se suma un vector adicional denominado Positional encoding el cual se encarga de incluir la información posicional referente al orden de los datos de entrada. El resultado de la suma de los vectores ingresa al Mecanismo de autoatención multi-cabeza, el cual, como su nombre lo indica, permite que cada dato preste atención a los demás datos utilizando varia cabezas para así capturar diferentes tipos de relación. A la salida, se obtiene una suma entre el mecanismo de autoatención con la entrada original y se aplica normalización. Finalmente, cada uno de los datos ingresa a una red neuronal "Feed forward" para dar ajuste final a los datos procesados.

Por otro lado, el decodificador es la parte encargada de generar la salida utilizando

la información procesada por el codificador. Su funcionamiento se basa en una serie de pasos que combinan la atención a los datos de entrada generados por el codificador con el procesamiento de los embeddings correspondientes a los tokens que ya han sido generados en la salida esperada. El decodificador recibe como entrada la secuencia objetivo desplazada. Esto significa que el modelo recibe la misma secuencia que debe predecir, pero desplazada hacia la derecha. Simultáneamente, se recibe el contexto generado por el codificador. Estos datos contienen información importante sobre las relaciones y patrones detectados. A lo largo del decodificador, esta información se combina con la secuencia objetivo desplazada. El decodificador también cuenta con mecanismo de autoatención y una red feed-forward, además de utilizar un mecanismo de atención cruzada. Primero, el mecanismo de autoatención se aplica exclusivamente a la secuencia objetivo desplazada, permitiendo que cada dato de la secuencia preste atención únicamente a los datos previos. Luego, el decodificador aplica el mecanismo de atención cruzada, que conecta las representaciones del codificador con las de la secuencia objetivo. Finalmente, cada token procesado pasa por una red feed-forward, que realiza una transformación adicional para ajustar las representaciones convertidas en los pasos anteriores. Luego de todo el proceso en el decodificador, la salida se procesa mediante una capa lineal y una función softmax, generando las probabilidades de los posibles datos de salida. [39, 44, 47]

2.5.6. Patch Time Series Transformer (PatchTST):

Recientemente, se ha comenzado a explorar el potencial de los transformers para trabajar con series de tiempo. Aunque originalmente los transformers fueron diseñados para manejar secuencias de texto, su arquitectura basada en la atención ha demostrado ser eficaz también para capturar patrones y dependencias temporales en datos secuenciales. PatchTST es un modelo descrito en [48] con la innovación de dividir los datos de entrada en patches de datos para utilizarlos de forma independiente y alimentar un modelo de transformer estándar.

El funcionamiento se centra en dos componentes clave: el parchado de las series temporales y la independencia de canales. El proceso principal se centra en el procesamiento de los datos de entrada (o las series temporales), estas se normalizan y se dividen en "patches", lo que permite agrupar pasos de tiempo locales que pueden contener valores similares. Este proceso no solo reduce la longitud de los datos de entrada (mejorando así la eficiencia computacional), sino también permite al modelo mantener una ventana de retroceso más larga, esencial para capturar patrones temporales complejos.

Posteriormente se realiza la metodología de un transformer convencional, aplicando la transformación de los datos en embeddings, la decodificación y demás pasos.

En la Figura 2.12, se presentan las dos versiones de la arquitectura del modelo PatchTST. Puede apreciarse que se presentan dos alternativas de uso, con diferencia de que, en el modelo Self-supervised se eliminan aleatoriamente partes de la entrada y el modelo se entrena para recuperar estos valores faltantes, permitiendo al modelo aprender representaciones útiles a partir de datos no etiquetados. [48]

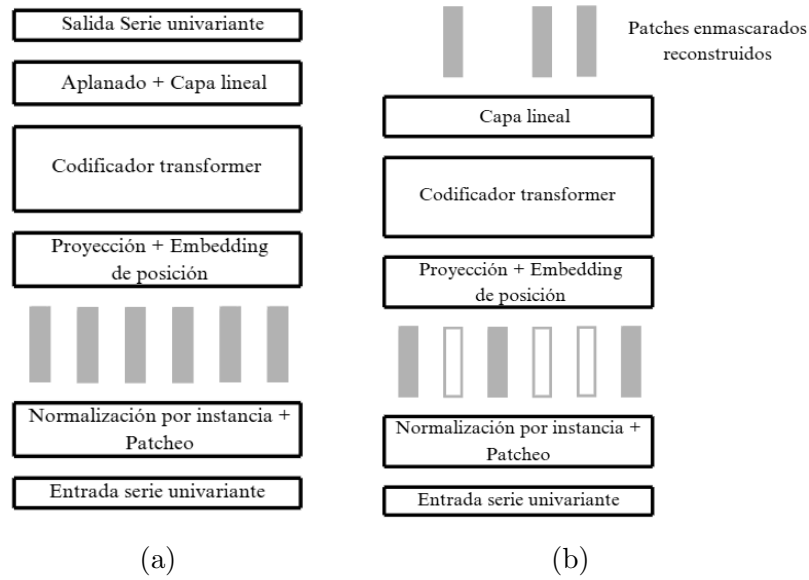


Figura 2.12: Arquitecturas PatchTST: (a) Supervisado, y (b) Auto-supervisado. Adaptado de [48].

Cuando los datos temporales son segmentados y alimentados a un modelo, este se enfoca en periodos cortos de tiempo, lo que repercute directamente en cómo aprende. La segmentación de la secuencia temporal en trozos más pequeños permite que el modelo se especialice en identificar patrones y relaciones dentro de esas ventanas de tiempo, mejorando su capacidad para detectar características locales y dinámicas dentro de la serie temporal. Sin embargo, este enfoque también presenta desafíos, ya que puede dificultar la captura de dependencias a largo plazo que abarcan más allá de los límites de cada segmento.

Cada uno de los modelos de DL ofrece características y ventajas particulares. Sin embargo, independientemente del tipo de modelo seleccionado, todos comparten la necesidad de realizar un preprocesamiento de los datos para maximizar el rendimiento del modelo. Este paso es crucial, ya que los modelos de DL requieren que los datos sean transformados de manera que puedan ser eficientemente interpretados y utilizados durante el proceso de entrenamiento. Gracias al preprocesamiento podemos filtrar, amplificar y segmentar la señal con la finalidad de clasificar las señales de una forma más concreta. [49]

2.6. Procesamiento de la señal EMG

La señal EMG se representa como una función del tiempo, lo que la hace compleja de interpretar directamente. Sin embargo, también puede describirse en términos de amplitud, frecuencia y fase. [50] Para lograr este objetivo, es necesario aplicar técnicas que reduzcan las dificultades al trabajar con señales EMG, facilitando así su análisis y, en última instancia, permitiendo una clasificación más precisa. [51] Además de la aplicación

de técnicas como el filtrado y la eliminación de señales no deseadas permitiendo así que la actividad muscular sea más clara y comprensible. [52]

2.7. Ruido en una señal EMG

Existen diversas fuentes que pueden afectar la calidad de la señal EMG. Entre los principales contaminantes se encuentran el ruido de línea base, el ruido de interferencia y los artefactos relacionados con las características fisiológicas y anatómicas de cada usuario. [53]

El ruido de la línea base es aquél presente en el momento en el que el músculo se encuentra en reposo, es decir, en el momento en el que no se genera ningún movimiento. [54] Este tipo de ruido es causado por la combinación de factores fisiológicos como la condición de la piel, pequeños movimientos involuntarios o la calidad de los electrodos utilizados. Sin embargo, las interferencias también pueden proceder de otro tipo de señales fisiológicas como la señal electrocardiográfica llegando a afectar en un espectro de frecuencia de hasta 100 Hz. [55] La diafonía de otro músculo, afectando directamente en el origen de la señal, pues cuando se presenta suele confundirse la señal de interés con la señal proveniente de otro músculo. O el ruido ambiental, proveniente de otros equipos como cables eléctricos, televisores, etc. [56]

Sin embargo, la fuente de ruido más problemática al trabajar con señales EMG es el ruido de interferencia, también conocido como interferencia de la línea de energía (PLI). Este tipo de ruido es causado por la presencia de oscilaciones electromagnéticas generadas por la red eléctrica, afectando los cables y dispositivos de registro. Dado que el PLI se origina en la red eléctrica, su frecuencia típica es de 60 Hz. [57] Otra fuente importante de ruido a considerar es el movimiento corporal durante el registro, el cual puede ser involuntario o generado por pequeños movimientos voluntarios durante la adquisición de la señal. [58] Generalmente, este tipo de ruido se presenta en frecuencias cercanas a 50 Hz, aunque su magnitud y características dependen del contacto entre el electrodo o sensor y la piel del usuario. [59]

2.8. Reducción de ruido en la señal EMG

Cada fuente de ruido posee diferentes características. Por lo que, afectan de forma diferente a la señal EMG. La frecuencia de una señal EMG oscila entre 10 y 500 Hz y posee una amplitud de entre 0 y 10 mV. [60]

La selección de la metodología a utilizar también depende del tipo de ruido que afecte a nuestra señal, a continuación se describen algunas metodologías empleadas para reducir el ruido presente al trabajar con señales EMG.

2.9. Filtros digitales

Los filtros digitales son especialmente diseñados para modificar o mejorar las características de una señal mediante operaciones matemáticas en el dominio digital. Su

objetivo principal es permitir el paso de componentes deseados de una señal mientras se atenúan o eliminan otras. [61]

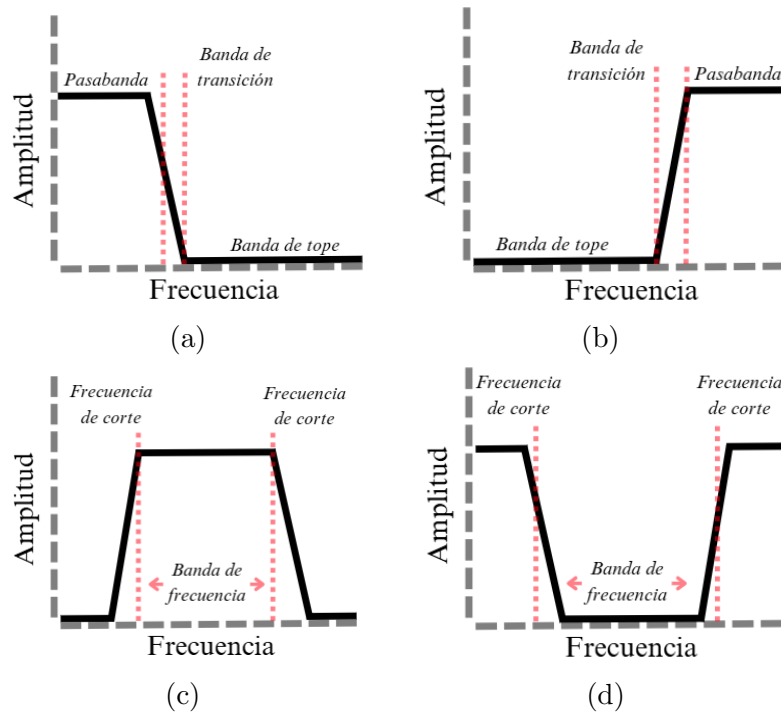


Figura 2.13: Cuatro filtros digitales más comunes: (a) Filtro pasa bajas, (b) Filtro pasa altas, (c) Filtro pasa bandas, y (d) Filtro rechaza banda. Adaptado de [62].

La figura 2.13 describe los cuatro filtros digitales más utilizados en la tarea de limpieza de señales: (a) filtro de paso bajo, (b) filtro de paso alto, (c) filtro pasa banda y (d) filtro rechaza banda. [63]

Para el uso de los filtros, es primordial considerar un umbral denominado como frecuencia (o frecuencias) de corte, dicho umbral nos permite controlar el paso de las frecuencias de nuestro interés, por ejemplo, el filtro de paso bajo es utilizado para atenuar las frecuencias presentes por encima de la frecuencia de corte. Es decir, permite pasar las frecuencias que se encuentran por debajo de dicha frecuencia de corte, mientras que atenúa o elimina las frecuencias superiores a esta. Este filtro es útil en aplicaciones donde se desea eliminar el ruido de alta frecuencia o suavizar señales, permitiendo solo el paso de componentes de frecuencia baja.

El filtro de paso alto es similar, con la diferencia de que permite el paso de las frecuencias que se encuentran por encima de la frecuencia de corte y atenúa aquellas inferiores. De esta forma, permite el paso a componentes de frecuencia alta.

Los filtros pasa banda y rechaza banda son una combinación de los filtros de paso bajo y de paso alto, ya que el filtro pasa banda permite el paso de un rango de frecuencias intermedias, es decir, aquellas que se encuentran entre dos frecuencias de corte, mientras

que atenúa las frecuencias inferiores y superiores a este rango. Por otro lado, el filtro rechaza banda, elimina un rango específico de frecuencias, permitiendo el paso de todas las demás, y es utilizado, por ejemplo, para suprimir interferencias en una frecuencia determinada. [63, 64]

2.9.1. Transformada de Fourier (FT)

Al trabajar con señales en series de tiempo es complicado comprender las frecuencias que afectan a la señal. La FT es una herramienta matemática que permite realizar una conversión de la señal en términos de frecuencia. [65]

Para ello, la FT expresa una señal como una suma de ondas sinusoidales, es decir, en términos de ondas senoidales y cosenoidales. Cada onda con una frecuencia, amplitud y fase específicas. Posteriormente, se utiliza la FT inversa para volver al dominio del tiempo.

La FT tiene distintas versiones, una de ellas es la transformada de Fourier discreta (DFT), que como su nombre indica, es útil para señales que contienen datos discretos o muestreados, es decir, señales representadas por un número finito de puntos.

Las señales EMG son continuas por naturaleza. Sin embargo, al ser adquiridas y procesadas, estas señales se convierten en datos muestreados (es decir, discretos) para su análisis digital o para distintas aplicaciones.

En la ecuación (4) se expresa matemáticamente la DFT y la ecuación (5) expresa la DFT inversa.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{kn} \quad (4)$$

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) \cdot W_N^{-kn} \quad (5)$$

Siendo, $W_N = e^{-j(\frac{2\pi}{N})}$ el factor base de la DFT, es decir, es el elemento que permite calcular los coeficientes de frecuencia de la señal. N , es la longitud de $[x(n)]$ (número total de muestras de la señal). [65]

Existe una versión optimizada y computacionalmente eficiente de la DFT llamada Transformada Rápida de Fourier (FFT). La FFT fue desarrollada para reducir significativamente el tiempo de cálculo necesario para realizar la DFT, especialmente cuando se trabaja con grandes cantidades de datos. [66] Para ello, la FFT trabaja con una técnica llamada "divide y vencerás", que divide la señal en segmentos más pequeños, calcula la DFT de los segmentos pequeños para finalmente combinar los resultados, obteniéndose así la DFT completa. [67]

2.9.2. Transformada Wavelet (WT)

La WT es un método de descomposición de señales propuesto como una alternativa a las limitaciones de la FT. La WT permite analizar una señal tanto en el dominio del

tiempo como en el dominio de la frecuencia simultáneamente, lo que la hace especialmente útil para las señales no estacionarias. [53]

Al igual que la FT, la metodología de la WT se divide en dos enfoques principales: la Transformada Wavelet Continua (CWT) y la Transformada Wavelet Discreta (DWT). [68]

A diferencia de la FT, que trabaja con ondas sinusoidales, la WT utiliza una función base llamada wavelet madre. Esta función base se escala y se traslada la señal EMG para generar distintas representaciones, lo que permite identificar las partes relevantes de la señal y separar el ruido de los componentes esenciales.

En este contexto, la CWT descompone la señal en una familia continua de frecuencias y posiciones en el tiempo, proporcionando una representación detallada tanto en el dominio temporal como frecuencial a diferentes escalas. La señal se analiza en segmentos locales mediante la wavelet madre, que se traslada utilizando las diferentes escalas. Por otro lado, la DWT realiza una descomposición discreta de la señal utilizando una discretización de las frecuencias y posiciones en el tiempo. Este proceso se lleva a cabo en niveles, dividiendo la señal en subbandas con diferentes resoluciones en tiempo y frecuencia. [69]

La selección de la wavelet madre depende principalmente de las características específicas de la señal que se desea analizar. La Tabla 2.2 presenta las 324 funciones wavelet, agrupadas en 15 familias.










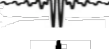

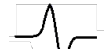
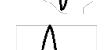


La ecuación (6) describe la ecuación de la CWT, que describe cómo se descompone una señal utilizando la wavelet madre de nuestra elección.

$$Wf(a, b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \quad (6)$$

La wavelet madre es escalada por el factor a , cuando $a > 1$, la wavelet madre se estira, suavizando la señal, permitiendo captar información correspondiente a los componentes de baja frecuencia, los cuales están asociados con estructuras más amplias y lentas de la señal. Por otro lado, cuando $a < 1$, la wavelet madre se comprime, volviéndose más nítida, enfocándose en los componentes de alta frecuencia, que corresponden a detalles más pequeños y rápidos de la señal. Este desplazamiento evoluciona a lo largo de la señal en un factor b , evaluando distintos puntos de la señal en el dominio del tiempo, analizando cómo las características de frecuencia varían en diferentes instantes. [70]

Es en este punto donde se analizan los factores obtenidos mediante la transformada wavelet, lo que permite detectar cambios bruscos innecesarios en la señal, generalmente interpretados como ruido. Para reducir este ruido, se aplican filtros de paso bajo o de paso alto, dependiendo de las características de la señal y de los objetivos del análisis. Posteriormente, la señal es reconstruida a partir de los datos procesados, conservando únicamente los componentes relevantes que corresponden a eventos significativos o "datos útiles" para el análisis posterior. Este proceso asegura que la señal final sea más limpia y adecuada para su interpretación.

Cuadro 2.2: Lista de la 324 wavelets madre ordenadas en 15 familias. Adaptado de [3]

Nombre wavelet	Subtipo Wavelet	Número wavelet	Figura
Haar	db1	1	
Daubechies	db2-db45	2-45	
Coiflet	coif1-coif5	46-50	
Morlet	morl	51	
Complex discrete morlet	cmor	52-147	
Discrete meyer	dmey	148	
Meyer	meyr	149	
Mexican hat	mexh	150	
Shannon	shan	151-200	
B-Spline	fbsp	201-260	
Gaussian	gaus	261-267	
Complex gaussian	cgaus	268-275	
Biorthogonal	bior	276-290	
Inverse biorthogonal	rbio	291-305	
Symlet	sym	306-324	

2.10. Segmentación de la señal EMG

La segmentación de señales es el proceso de dividir una señal en partes más pequeñas, conocidas como segmentos, con el fin de analizar características específicas de la señal en intervalos de tiempo determinados. Permitiendo estudiar comportamientos locales y facilitando la identificación de patrones o eventos particulares dentro de la señal. [71, 72]

2.10.1. Segmentación por eventos

La segmentación por eventos consiste en dividir una señal o un conjunto de datos en partes o segmentos específicos que corresponden a eventos de interés previamente definidos. [73] Un evento se define como un fenómeno o característica particular en los datos que cumple ciertos criterios, como umbrales de amplitud, cambios en la frecuencia, o la ocurrencia de patrones temporales. En el caso de señales mioeléctricas, estos eventos suelen corresponder a contracciones musculares o movimientos específicos.

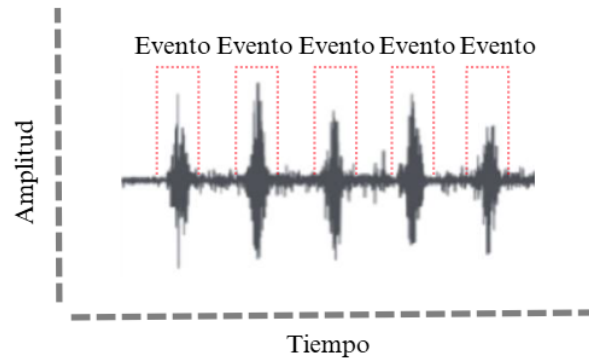


Figura 2.14: Segmentación mediante inicio y final de un evento muscular. Adaptado de [74]

La figura 2.14 presenta la segmentación por eventos, donde se tiene el conocimiento del inicio y fin de las contracciones musculares, de esta forma puede despreciarse aquellos datos que correspondan a tiempos de relajación muscular o tiempos muertos que no aportan significancia a la señal.

Esta técnica es especialmente útil cuando se tiene conocimiento previo sobre el inicio de la contracción o el momento exacto en el que se ejecutó un movimiento, de esta forma nos aseguramos de extraer el evento y evitar datos irrelevantes o innecesarios.

2.10.2. Segmentación por ventanas deslizantes

En ocasiones, es necesario realizar una segmentación basada en intervalos cortos de tiempo. Para ello, se utiliza una técnica conocida como ventaneo. El ventaneo consiste en dividir la señal en segmentos temporales de duración fija, denominados ventanas, y posteriormente aplicarles un filtro para suavizar los bordes de los segmentos y minimizar los efectos de las discontinuidades, como la ventana de Hamming o Hanning. [75]

Sin embargo, el ventaneo también puede aplicarse sin necesidad de un filtro, limitándose a la metodología de segmentación para realizar el análisis posterior de los datos.

La figura 2.15, describe las dos metodologías para aplicar el ventaneo: ventaneo adyacente y ventaneo superpuesto. En el ventaneo adyacente, se toma una longitud predefinida de ventanas, y la siguiente ventana comienza al final de la ventana precedente

hasta completar el total de datos de la señal. En el ventaneo superpuesto, se toma una longitud de ventana predefinida, que se desliza sobre la señal con un paso o incremento inferior al tamaño de la ventana. [76]

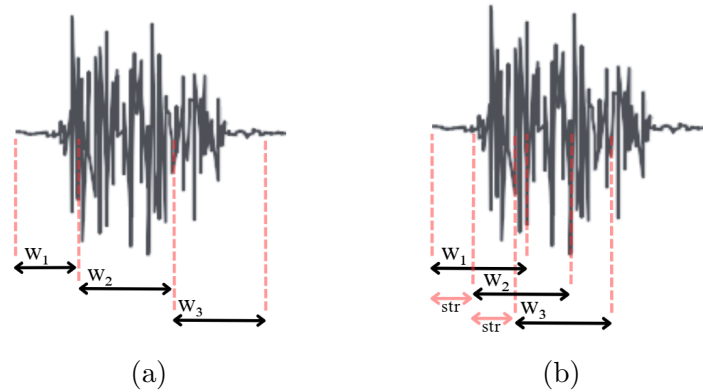


Figura 2.15: Métodos de segmentación por ventanas: (a) Ventaneo adyacente, y (b) Ventaneo superpuesto. Adaptado de [76, 77].

Para realizar el ventaneo es necesario considerar un paso, este valor nos permitirá desplazar la ventana a la derecha y así recorrer la señal completa. La ecuación (7) describe el desplazamiento de la ventana a lo largo de la señal. [78]

$$x(t + S) = \{x(t + S), x(t + S + 1), \dots, x(t + S + W - 1)\} \quad (7)$$

donde W , representa el tamaño de la ventana y Str el paso o desplazamiento que realizar la ventana a la derecha.

Este tipo de técnicas de ventaneo son útiles cuando se requiere obtener un análisis más detallado sobre los cambios que ocurren en periodos cortos de tiempo en la señal. Cada ventana permite aislar un segmento temporal específico, enfocándose en diferentes periodos de tiempo. Posteriormente, los puntos dentro de cada ventana pueden ser analizados de manera individual o comparativa para identificar patrones o características relevantes.

2.11. Extracción de características

Después de haber procesado de la señal EMG, es necesario caracterizarla, pues apesar de que los datos de interés se encuentran implícitos en la señal, es necesario aplicar metodologías que permitan el análisis y extracción de estos. [79]

La caracterización permite extraer información relevante, como las propiedades temporales, frecuenciales o estadísticas, que describen de manera más efectiva el comportamiento de la señal y facilitan su interpretación o clasificación en etapas posteriores.

Dentro de la extracción de características, se incluyen diversas técnicas tanto manuales como automáticas. Las técnicas manuales implican el uso de métodos tradicionales para

identificar y extraer características relevantes de la señal. Estas características suelen ser seleccionadas basándose en el conocimiento previo de la señal y los patrones que se desean identificar, como por ejemplo el valor absoluto medio (MAV), error cuadrático medio (MSR), varianza (VAR), entre otros. Este tipo de análisis ayudan a proporcionar una descripción útil de la señal. [7]

Sin embargo, existen técnicas automáticas para la extracción de características, permitiendo extraer características sin intervención manual.

Las metodologías automáticas tienen la capacidad de aprender de los datos de manera profunda, descubriendo representaciones complejas de la señal EMG que podrían no ser evidentes mediante métodos tradicionales. [80]

La extracción de características es de suma importancia, ya que constituye la base para lograr una clasificación precisa. [81]

2.11.1. Extracción de características utilizando CNN

La estructura de una CNN está diseñada específicamente para identificar y analizar patrones, lo que la convierte en una herramienta ideal para el procesamiento de imágenes. [82]

Sin embargo, gracias a su diseño, una CNN también puede aplicarse en diferentes enfoques, ya que es capaz de procesar datos estructurados en forma de cuadrícula, como los píxeles en imágenes, las cuales son posibles de representar como una matriz de características. Esta matriz actúa como una entrada para la CNN, permitiendo que la red extraiga patrones espaciales y temporales relevantes para la clasificación de señales EMG.

Cuando los datos temporales son segmentados, cada segmento se trata como un conjunto de datos independiente.

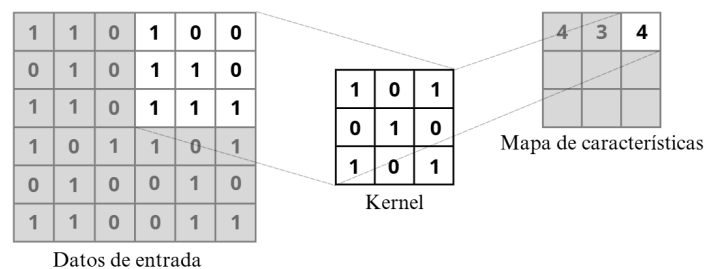


Figura 2.16: Extracción de características in una CNN. Adaptado de [83]

En la figura 2.16, se describe el proceso de caracterización de la señal mediante la aplicación de un filtro o kernel a los datos de entrada. Para realizar la extracción de características, la CNN realiza una operación llamada convolución, la cual realiza una multiplicación elemento a elemento entre un filtro (llamado kernel) y una región específica de los datos de entrada, seguida de una suma de los resultados obtenidos, la ecuación (8) describe cómo se realiza la convolución.

$$S(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i + m, j + n) \cdot K(m, n) \quad (8)$$

La operación de convolución permite recorrer la matriz de entrada a través de las filas y columnas. En cada posición, se realiza la convolución entre el kernel y los valores de la entrada en esa región local, lo que da como resultado un único valor que se almacena en el mapa de características. Este proceso se repite mientras el kernel se desliza sobre toda la matriz de entrada.

A continuación, se aplica una función de activación, generalmente la unidad lineal rectificadora (ReLU), que introduce no linealidad en el modelo. Con esto, se asegura que la red pueda aprender relaciones complejas entre los datos mediante la eliminación de los valores negativos que podrían haber sido generados durante la convolución.

Finalmente, se lleva a cabo una operación de reducción de dimensionalidad mediante la capa de pooling. Este paso disminuye la resolución espacial del mapa de características, ayudando a reducir el tiempo de cálculo y a controlar el overfitting, mientras conserva las características más importantes de la señal. Este proceso se repite a través de múltiples capas para extraer características jerárquicas de mayor nivel.

2.11.2. Extracción de características utilizando Transformers

Previamente se ha descrito la segmentación de la señal mediante técnicas de ventaneo deslizante. Estas segmentaciones se trabajan como tokens una vez ingresando a la metodología de un transformers. Los tokens son representaciones segmentadas de la señal completa [44], dichos tokens permiten aprender dependencias dentro de periodos cortos de tiempo.

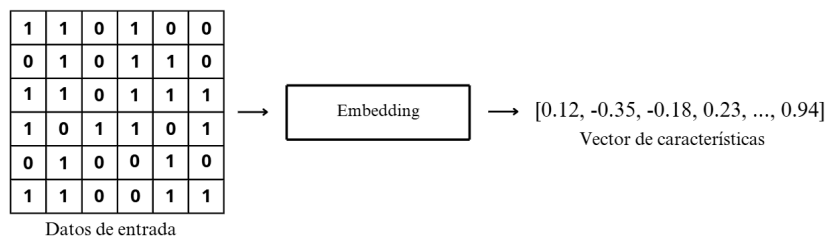


Figura 2.17: Extracción de características en un transformer. Adaptado de [44]

La figura 2.17, describe como los datos procesados referentes a la señal EMG sirven como entrada para realizar la caracterización mediante el proceso de embedding del transformer. La ecuación (9) describe la generación del vector de características.

$$z_i = W_e x_i + b_e \quad (9)$$

Donde cada una de las ventanas es multiplicada por una matriz de pesos, dicha matriz es la que se ajusta en cada iteración durante el aprendizaje del modelo. De manera inicial, estos pesos se inicializan con valores aleatorios. Además, es necesario considerar el positional encoding, el cual no solo permite aprender los datos correspondientes a la ventana, sino también su posición en la secuencia. Esto es fundamental para capturar cómo evolucionan los datos a lo largo del tiempo.

El positional encoding genera etiquetas únicas para cada posición en la secuencia, utilizando funciones seno y coseno, lo que permite representar de manera diferenciada cada ventana en la secuencia, asegurando que el modelo pueda distinguir y relacionar diferentes ventanas en el tiempo. Las ecuaciones (10) y (11) describen el cálculo del positional encoding.

$$PE(i, 2k) = \sin\left(\frac{i}{10000^{\frac{2k}{d_{\text{embed}}}}}\right) \quad (10)$$

$$PE(i, 2k + 1) = \cos\left(\frac{i}{10000^{\frac{2k}{d_{\text{embed}}}}}\right) \quad (11)$$

Finalmente, utilizando la ecuación (12), se genera la entrada al modelo transformer, que utiliza esta información para resolver la tarea de clasificación de manera eficiente.

$$z_i = \mathbf{W}_e \mathbf{x}_i + \mathbf{b}_e + PE_i \quad (12)$$

Así, el vector de características final se obtiene al combinar el input embedding con el positional encoding, integrando tanto el contenido de la ventana como su posición en la secuencia. Esto asegura que la información sobre la evolución de la señal a lo largo del tiempo esté disponible para el modelo.

2.12. Clasificación

Un modelo de clasificación permite identificar patrones en las señales para interpretar los datos de entrada. Un clasificador debe tener una capacidad de generalización, lo que le permite realizar predicciones precisas sobre datos no vistos o nuevos. Además, debe ser capaz de trabajar con datos multidimensionales, lo que es esencial para abordar problemas complejos como el análisis de señales, y debe ser robusto, es decir, capaz de mantener su rendimiento incluso en presencia de ruido, datos incompletos o variabilidad en las características de entrada. [7]

Existen diversos tipos de clasificadores basados en IA. Entre ellos, los modelos de DL se destacan como una herramienta poderosa para realizar la clasificación de señales de manera robusta y eficiente, capturando patrones complejos y características intrínsecas de los datos.

La combinación de modelos para la extracción de características y el reconocimiento

de patrones suele ser una de las alternativas más completas en cuanto a clasificadores, ya que integra la capacidad de extraer características de manera automatizada con el aprendizaje basado en secuencias.

Una estrategia ampliamente utilizada es la combinación de redes CNN para la extracción de características y LSTM para el aprendizaje de las dependencias temporales presentes en las características extraídas. Por otro lado, los modelos transformer han surgido como una alternativa prometedora a las redes RNN, ya que son capaces de manejar de manera más eficiente las dependencias a largo plazo en datos secuenciales.

Cuando se trabaja con señales EMG, el clasificador debe ser capaz de interpretar datos nuevos para analizar las secuencias temporales o las características extraídas de las señales, con el objetivo de identificar el tipo de movimiento al que pertenecen. Esto representa uno de los desafíos más complejos en el diseño y entrenamiento de clasificadores, debido a la naturaleza ruidosa y variable de las señales EMG.

3. Metodología

A continuación se presenta la metodología propuesta. Considerando lo aprendido a lo largo de la investigación. La figura 3.18, describe el flujo de trabajo aplicado.

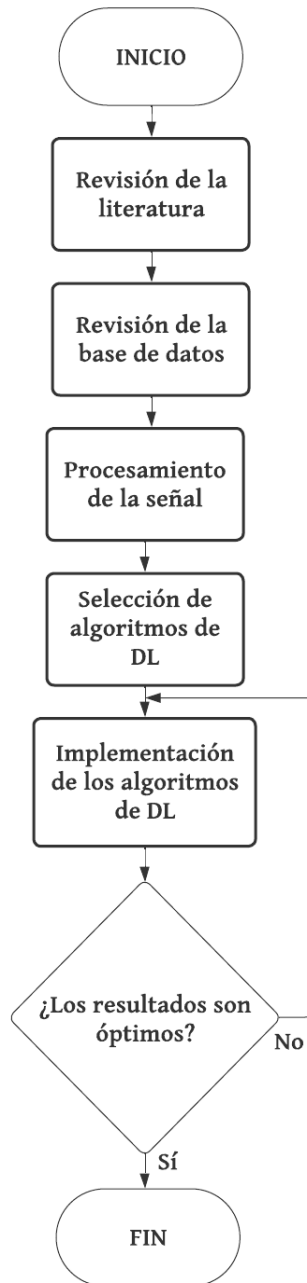


Figura 3.18: Diagrama de flujo de la metodología.

Antes de presentar la base de datos utilizada para la realización del proyecto, es necesario considerar los músculos participes en la ejecución de los movimientos. Para ello, la tabla 3.3 proporciona una descripción acerca de los músculos involucrados y qué importancia tienen en el movimiento.

Cuadro 3.3: Descripción de cada músculo implicado en los movimientos. [84]

Músculo	Descripción
Tríceps	Es un músculo grande situado en la parte posterior del brazo. Se compone de tres cabezas: larga, lateral y medial. La principal función del tríceps es la extensión del codo, que permite estirar el brazo.
Extensores	Son un grupo de músculos situados en la parte posterior del antebrazo. Su función principal es extender las muñecas, los dedos y, en algunos casos, el codo.
Supinador	Es un músculo del antebrazo encargado de realizar el movimiento de supinación, es decir, girar la palma de la mano hacia arriba o hacia delante.
Extensores del carpo	Son un grupo de músculos del antebrazo que se encargan de extender la muñeca, es decir, de moverla hacia atrás. El grupo de músculos está formado por: Carpi radialis longus, Carpi radialis brevis y Carpi ulnaris.
Braquioradial	Es un músculo del antebrazo situado en la parte lateral del antebrazo, y desempeña un papel importante en los movimientos del codo, participa en la pronación y supinación del antebrazo.
Braquial	Es un músculo del brazo situado en la parte anterior, justo debajo del bíceps. Es uno de los principales flexores del codo y desempeña un papel esencial en los movimientos de flexión de la articulación del codo.
Flexores	Son los músculos que permiten la flexión del codo, es decir, el movimiento que disminuye el ángulo entre el brazo y el antebrazo. Siendo un conjunto de músculos entre el braquial y el carpo radial.
Pronador	Es un músculo que interviene en el movimiento de pronación del antebrazo. Al girar el antebrazo de modo que la palma de la mano quede orientada hacia abajo o hacia atrás. Siendo una articulación entre el pronador redondo y el pronador cuadrado.
Flexores superficiales	Son un grupo de músculos del brazo y la mano que intervienen en el movimiento de flexión. Los flexores superficiales son esenciales para realizar acciones de flexión de la muñeca y los dedos.
Flexor profundo	Son los músculos más profundos del antebrazo responsables de la flexión articular. Es crucial para las actividades que requieren una flexión fuerte y profunda de los dedos.

3.1. Materiales

Es importante considerar los materiales a utilizar para conseguir nuestro objetivo. Con respecto al hardware utilizado, se empleo el equipo Dell Inspiron 15 7000 gaming, el cual cuenta con un procesador Intel(R) Core(TM) i5-7300HQ, con una velocidad de CPU a 2.50 GHz, con 16 GB de memoria RAM y una tarjeta de video NVIDIA GeForce GTX 1070. Sin embargo, debido a la alta capacidad computacional requerida para los entrenamientos, especialmente en modelos robustos como los transformers, se utilizó también la plataforma en línea Google Colab Pro, que ofrece acceso a diversas GPUs, específicamente optimizadas para el entrenamiento de modelos de IA, como la NVIDIA A100 o la Tesla T4.

Con respecto al software, se utilizó el lenguaje de programación Python, que es compatible con Google Colab. Además, Google Colab proporciona acceso fácil a diversas librerías de código abierto, como Numpy, Pandas, TensorFlow, entre otras.

3.2. Base de datos

Existen dataset públicos que permiten una clasificación idónea basada en señales EMG. Sin embargo, muchos de ellos no contemplan una población amplia, lo que repercute directamente en no tener la suficiente información para aprender la variabilidad entre sujetos. Además, una característica compartida en datasets públicos es que contienen datos listos para la clasificación, lo que afecta directamente en tener métricas altas sin necesidad de realizar un procesamiento previo.

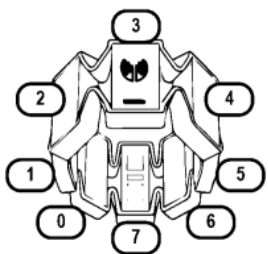
Para la clasificación, se utilizó un dataset privado creado en la Universidad Autónoma de Querétaro [85]. Los datos fueron obtenidos mediante el brazalete "Myo armband", desarrollado por Thalmic Labs. El Myo armband está equipado con sensores EMG que detectan los movimientos musculares, así como sensores inerciales para capturar movimientos del brazo. Representa una alternativa eficiente frente a las problemáticas asociadas a los electrodos de superficie tradicionales, ya que elimina completamente la necesidad de cableado para la extracción de las señales en los usuarios.

La base de datos fue diseñada con el objetivo de proporcionar una muestra representativa de diversas características demográficas para el análisis de las señales EMG. La distribución por género es equilibrada, con una ligera mayoría de mujeres (29) en comparación con los hombres (21), lo que permite examinar posibles diferencias de género en las señales. Además, los participantes fueron evaluados en condiciones óptimas, asegurándose de que no presentaran antecedentes de enfermedades neuromusculares, problemas motrices, ni se encontraran en estado de embarazo o lactancia.

En la tabla 3.4, se describe la posición de cada electrodo y su relación con respecto a la colocación en el brazo de cada sujeto.

Cada músculo involucrado está intrínsecamente relacionado con los demás. Sin embargo, no todos los músculos participan al mismo tiempo en la ejecución de cada movimiento individual.

Cuadro 3.4: Distribución de los sensores de la Myo armband en cada músculo. Adaptado de [85]

Myo armband	Sensor	Músculo
	0	Tricep
	1	Extensores y Supinador
	2	Extensores y Extensores del carpo
	3	Extensores y Braquioradial
	4	Braquial y Flexores
	5	Flexores y Pronador
	6	Flexores y Flexor superficial
	7	Flexores y Flexor profundo

En la figura 3.19 se presentan los movimientos realizados por cada participante. Cada movimiento se repitió por 5 veces en un intervalo de 3 segundos, es decir, por sujeto se toman 15 segundos de datos. Considerando también los tiempos de relajación, siendo estos la pausa entre eventos. Cada movimiento fue realizado manteniendo el codo pegado al torso y con instrucciones proporcionadas por fisioterapeutas.

- 3.19(a) Posición inicial: Mano en posición inicial, con la palma extendida y los dedos cerrados, sin ejercer demasiada fuerza. Los movimientos posteriores partirán de esta postura.
- 3.19(b) Pronación: Brazo cerca del tronco, el codo flexionado a 90° y la mano en posición inicial, la palma se gira hacia abajo y vuelve a la posición inicial.
- 3.19(c) Supinación: Brazo pegado al tronco, el codo en flexionado a 90° y la mano en posición inicial, la palma se gira hacia arriba y vuelve a la posición inicial.
- 3.19(d) Extensión: Brazo cerca del tronco, el codo flexionado a 90° y la mano en posición inicial. La palma se lleva hacia fuera sin mover el codo y se vuelve a la posición inicial.
- 3.19(e) Flexión: Brazo pegado al tronco, el codo flexionado a 90° y la mano en posición inicial. La palma se lleva hacia dentro sin mover el codo, de cara al cuerpo, y luego se vuelve a la posición inicial.
- 3.19(f) Desviación cubital: Brazo cerca del tronco, el codo flexionado a 90° , la mano en posición inicial y el brazo se mueve sin mover el codo, se dobla hacia abajo y vuelve a la posición inicial.
- 3.19(g) Desviación radial: Brazo cerca del tronco, el codo flexionado a 90° y la mano en posición inicial, la mano se inclina ligeramente hacia arriba sin mover el codo y vuelva a la posición inicial.

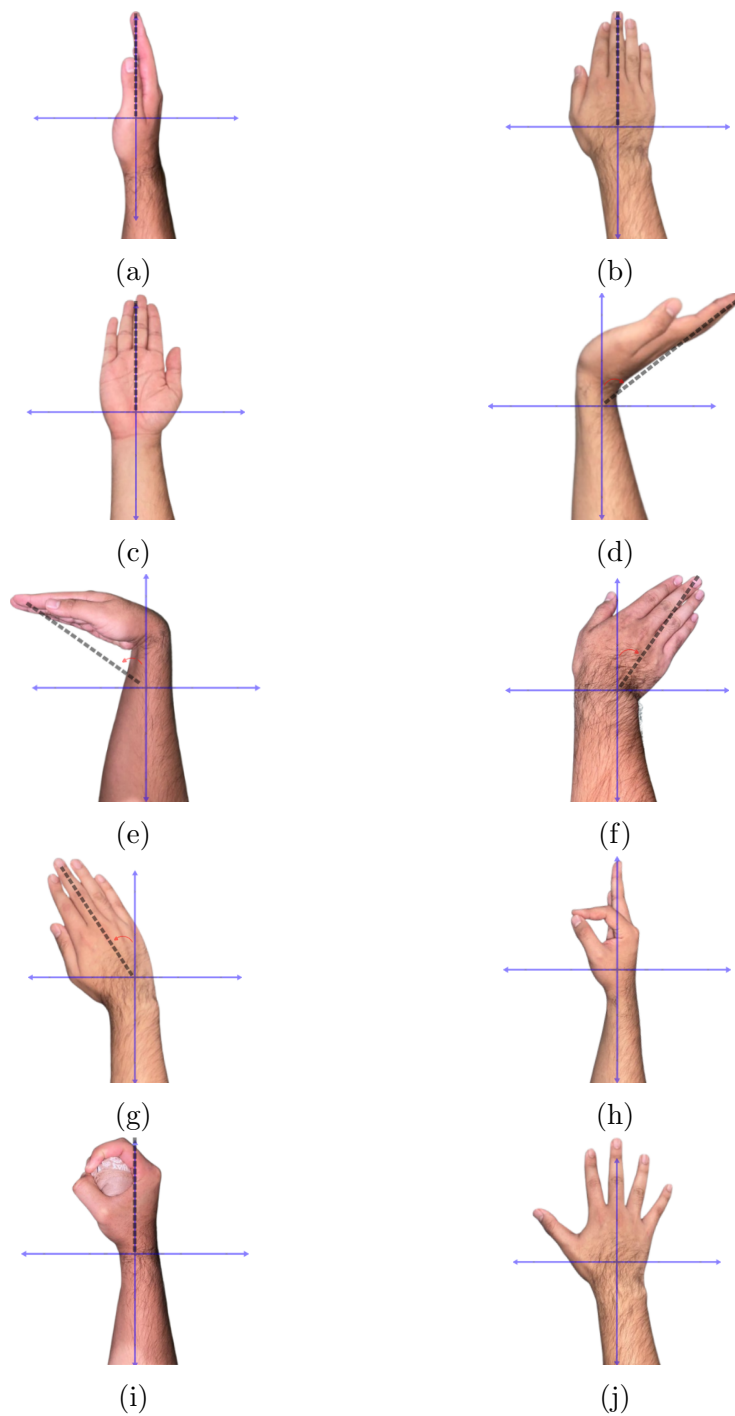


Figura 3.19: Lista de movimientos: (a) Posición inicial, (b) Pronación, (c) Supinación, (d) Extensión, (e) Flexión, (f) Desviación cubital, (g) Desviación radial, (h) Pinza fina, (i) Pinza gruesa, (j) Expansión. Adaptado de [85].

- 3.19(h) Pinza fina: Brazo cerca del tronco, el codo flexionado a 90° y la mano en posición inicial. Se tocan las yemas de los dedos pulgar e índice, y vuelve a la posición inicial.
- 3.19(i) Pinza gruesa: Brazo pegado al tronco, codo flexionado a 90° , mano en la posición inicial, cerrando el puño, dejando el pulgar fuera sobre los otros dedos, y se vuelve a la posición inicial.
- 3.19(j) Expansión: Brazo pegado al tronco, codo flexionado a 90° , mano en posición inicial, se separan todos los dedos y se vuelve a la posición inicial.

Para finalmente, en base a la posición del brazalete con respecto a los músculos participantes, se presenta la tabla 3.5, que relaciona los sensores participantes en cada movimiento.

Cuadro 3.5: Sensores involucrados en los movimientos. Adaptado de [87]

Movimiento	Sensores involucrados
Posición inicial	No involucrados
Pronación	2, 3 y 6
Supinación	2 y 3
Extensión	2, 3 y 4
Flexión	0, 6 y 7
Desviación cubital	0, 1 y 2
Desviación radial	3, 4 y 5
Pinza fina	1 y 2
Pinza gruesa	0, 1, 5 y 7
Expansión	0 y 3

No todos los sensores son participes en los movimientos generados, sin embargo, todos fueron considerados al momento de la clasificación para no perder ningún tipo de característica que se encuentre implícita en todos los sensores.

Como se describió en secciones anteriores, al momento que la señal es extraída se presenta el ruido de interferencia, este tipo de ruido siempre se encuentra presente. Por lo que, para mitigarlo se aplicó un filtro band-reject, específicamente un filtro notch, al momento que los sujetos ejecutaban el movimiento.

Además del brazalete Myo armband, se utilizó el software EMGSR, desarrollado en la Universidad Autónoma de Querétaro. La interfaz enseña el movimiento a los participantes y posteriormente realiza la recolección de los datos generados por la Myo armband, aplica el filtrado para eliminar el ruido de interferencia al momento de la obtención de la señal y almacenando los datos en un archivo .csv para su posterior uso.

3.3. Procesamiento de la señal

Generalmente, cuando se trabaja con señales bioeléctricas, el proceso de extracción no nos produce una señal completamente ideal para trabajar en metodologías que involucren la clasificación de datos.

Es por eso que el procesamiento de la señal es fundamental para extraer información relevante, reducir el ruido y mejorar la interpretación de los datos. [86]

En las siguientes secciones se describe el procesamiento realizado sobre los datos para mejorar la calidad de la señal después de haber sido adquirida de los sujetos.

3.4. Eliminación de ruido

A pesar de aplicar un filtrado durante la extracción de la señal, este no es capaz de eliminar completamente los ruidos producidos por movimientos involuntarios, músculos adyacentes, entre otros. Por ello, es necesario aplicar un filtrado posterior a la adquisición, lo que permite trabajar con la mayor cantidad posible de datos significativos de la señal.

Como mencionamos previamente, existen distintas metodologías para la eliminación de ruido en señales. Sin embargo, dentro del estado del arte, las WT han destacado debido a su capacidad para descomponer la señal en diferentes bandas de frecuencia y tiempo simultáneamente. Esta propiedad es particularmente útil para la limpieza de ruido mediante una técnica conocida como wavelet denoising, que utiliza la WT para reducir o eliminar el ruido presente en la señal, preservando sus características esenciales. [87]

Es necesario seleccionar la wavelet madre que más se acople a las características de nuestra señal, pues, de acuerdo a lo visto previamente, la wavelet madre es la encargada de realizar el procesamiento con respecto a nuestra señal original. Para nuestro interés, se seleccionó la Daubechies 20 (db20), lo que significa que a lo largo de la señal el análisis se realizará con respecto a una ventana con 20 puntos de soporte a lo largo de toda la señal. Además, se seleccionó un nivel de descomposición igual a 5, lo que indica que la wavelet madre será analizada en 5 niveles diferentes.



Figura 3.20: Comparación entre la wavelet madre db20 y la señal EMG original.

La figura 3.20 muestra la comparación a realizar entre la wavelet madre y la señal a

procesar. Para ello, tomando la ventana de puntos, se realiza una descomposición en dos coeficientes: coeficiente de aproximación y coeficiente de detalle. [88] Los cuales serán extraídos en el desplazamiento de la wavelet db20 a lo largo de la señal EMG de cada sujeto.

Los coeficientes de aproximación corresponden a la parte de la señal que captura la información de baja frecuencia, representando las características más suaves de la señal, para nuestro interés, se traduce a los eventos presentes en la señal. Por otro lado, los coeficientes de detalle representan las variaciones rápidas o los cambios bruscos, es decir, la información de alta frecuencia, considerándose como el ruido que nos interesa atenuar. A pesar de que la señal EMG es una señal continua, para la limpieza de ruido en la señal se hace uso de la DWT en cada proceso de comparación.

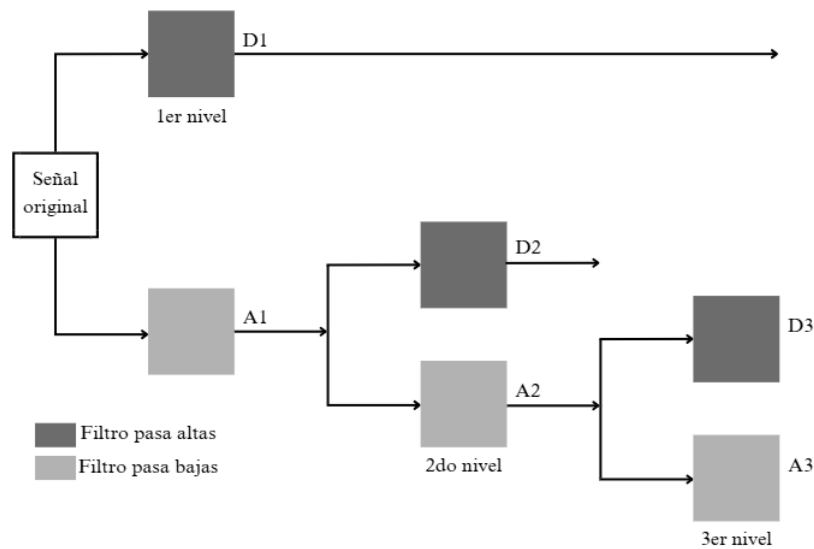


Figura 3.21: Descomposición en niveles mediante DWT. Adaptado de [89]

La figura 3.21 demuestra cómo se realiza la descomposición de la señal al momento de realizar el análisis con la wavelet madre. Se realizó una descomposición de 3 niveles de descomposición. Cada nivel de descomposición aplica los filtros pasa baja y pasa alta de manera jerárquica, dividiendo la señal en componentes más específicos. En cada nivel de descomposición, la señal se separa en un componente de aproximación (A_n) y un componente de detalle (D_n). Este proceso se repite para las aproximaciones, generando componentes de mayor nivel, es decir, se realiza una descomposición sucesiva de los coeficientes de aproximación en cada nuevo nivel, mientras que los coeficientes de detalle se mantienen en su nivel correspondiente.

Cada uno de los niveles contiene información diferente sobre la señal, por lo que de esta forma podemos analizar la señal a diferentes escalas. También es importante mencionar que, en cada uno de los niveles para la limpieza o atenuación del ruido se aplica una umbralización buscando que aquellos valores por debajo del umbral se

atenuan y los que superen el umbral se mantengan. Para nuestro caso de interés se aplicó una umbralización suave con valor de 30, buscando que aquellos valores que se buscan atenuar se suavicen si son menores que el umbral y los que son mayores permanezcan. Además de realizar un ligero ajuste a los datos de nuestro interés.

Posterior a la aplicación de los filtros en cada nivel, se realiza la reconstrucción de la señal utilizando la ecuación (13), que describe el principio de la Transformada Inversa de Wavelet Discreta (iDWT). Combinándose los coeficientes de aproximación y detalle obtenidos en cada nivel, junto con los filtros correspondientes, para recuperar la señal original procesada.

$$x(n) = \sum_{j,k} c_{j,k} \phi_{j,k}(n) + \sum_{j,k} d_{j,k} \psi_{j,k}(n) \quad (13)$$

Dentro de la ecuación se realiza la suma de los valores obtenidos para aproximación y detalle. Donde $c_{j,k}$ son los coeficientes de aproximación, que capturan la información de baja frecuencia de la señal en el nivel de descomposición j y posición k , asociados a los eventos de interés presentes en la señal, $\phi_{j,k}(n)$ es la función de escalamiento asociada a estos coeficientes de aproximación, utilizada para reconstruir dichas características relevantes. Y los valores obtenidos para los coeficientes de detalle, donde, $d_{j,k}$ son los coeficientes de detalle que contienen la información de alta frecuencia de la señal, es decir, el ruido o variaciones rápidas no deseadas. Finalmente, $\psi_{j,k}(n)$ es la función wavelet asociada a los coeficientes de detalle, utilizada para reconstruir estas variaciones rápidas en la señal. [90]



Figura 3.22: Comparación entre las señales del movimiento de flexión: (a) Señal bruta, y (b) Señal procesada.

La figura 3.22, muestra la comparación entre una señal del movimiento de flexión donde (a) muestra la señal sin procesar, y (b) la señal después del procesamiento aplicando la metodología con wavelet denoising.

El proceso se repitió para los 10 movimientos, donde también se realizó una exploración detallada de cada una de las señales para comparar el cambio producido con la metodología wavelet denoising.

Una vez terminada la fase de limpieza de ruido se procedió a realizar la segmentación de la señal pues como se mencionó en secciones anteriores, cada sujeto repitió el

movimiento cinco veces en diferentes intervalos de tiempo. Por lo que, para un mejor análisis cada repetición se utilizó como un segmento independiente.

3.5. Segmentación

La segmentación permite enfocar el análisis en detalles específicos que se presentan en intervalos de tiempo cortos dentro de la señal, los cuales podrían no ser evidentes al observar la señal completa.

El primer paso en el proceso de segmentación consiste en dividir cada registro utilizando la segmentación por eventos. Obteniendo pequeños segmentos para analizarlos de forma independiente y no como un único conjunto continuo. Esto es importante debido a la naturaleza de los datos registrados, que corresponden a señales completas de 16 segundos por sujeto. Durante este tiempo, cada sujeto repite el movimiento siguiendo tiempos específicos descritos en la Tabla 3.6, donde t_i representa el tiempo inicial del evento, y t_f el tiempo donde el evento termina.

Cuadro 3.6: Tiempos considerados para cada repetición del evento

Evento	t_i	t_f
1	0.5	3.5
2	3.5	6.5
3	6.5	9.5
4	9.5	12.5
5	12.5	15.5

Cada evento es segmentado de acuerdo con los tiempos de adquisición definidos, considerando un margen de 0.5 segundos entre eventos consecutivos. El margen se propone para aquellas posibles diferencias en los tiempos de adquisición, ya que, aunque los datos se registraron simultáneamente, pueden existir desajustes entre los tiempos de adquisición de los datos.



Figura 3.23: Segmentación de los eventos basados en el tiempo de recolección.

Al hacer la división por eventos, los datos para utilizar en la clasificación aumentan pues, apesar de que cada uno de los eventos segmetandos provienen del mismo sujeto y

Por movimiento	
Datos no segmentados	Datos segmentados
500 señales	2500 señales

Cuadro 3.7: Total de señales con datos no segmentados y segmentados.

pertenecen al mismo movimiento no son idénticos. La figura 3.23 muestra la segmentación realizada para cada señal, almacenando en archivos separados cada evento con su tiempo correspondiente.

Una vez que los datos son divididos en segmentos con respecto al tiempo de cada evento. Se presenta una verificación de datos, para tener la relación de los datos antes y después de la segmentación.

La tabla 3.7, muestra la comparación de los datos antes y después de la segmentación para cada uno de los movimientos.

Cabe aclarar que para cada uno de las señales se consideran 8 canales de lectura, pues recordando secciones anteriores, la lectura se realizó con la Myo armband, la cual cuenta con 8 sensores. Sin embargo, para el cálculo de las señales consideramos las 8 lecturas como una sola muestra.

El cálculo de la cantidad de señales se realizó con las ecuaciones (14) y (15), donde N_s representa el número de sujetos, N_m el número de movimientos y N_e representa el total de eventos presentes en la señal.

$$S = N_s \times N_m \tag{14}$$

$$S = N_s \times N_m \times N_e \tag{15}$$

La segmentación de los datos es completamente necesaria debido a la naturaleza de los datos. Sin embargo, esto no garantiza un correcto análisis para la clasificación. Es por ello que, se realiza una segmentación de los eventos aplicando la metodología de ventaneo superpuesto. De esta forma no solo aseguramos enfocarnos en periodos cortos de tiempo en la señal (lo que garantiza tener un control sobre lo que ocurre en distintos tiempos), si no también en incrementar la cantidad de datos para el entrenamiento. Además de aprender dependencias que no podrían ser analizadas si se ingresan las señales como una sola en el modelo. [91]

Para realizar la segmentación de la señal, es necesario definir de manera precisa tanto el tamaño de ventana como el desplazamiento (paso) con el cual se recorrerá la señal, para posteriormente calcular el número total de ventanas generadas para cada evento.

Se propone un tamaño de ventana de 400 muestras con un paso de 10 muestras. Recordando que para cada evento se considera un margen temporal con el fin de asegurar que los 3 segundos efectivos del evento sean completamente abarcados en el proceso de ventaneo. Adicionalmente, se lleva a cabo el análisis de la señal con el mayor número

de muestras, ya que es a partir de este segmento donde se establecen los parámetros óptimos para la segmentación (tamaño de ventana y paso), siendo este de 640 muestras.

La ecuación 16, describe el cálculo de las ventanas generadas para cada segmento. Después del cálculo realizamos un incremento con valor de 1 para asegurarnos que la mayor cantidad de datos sean contemplados en el ventaneo.

$$\text{Ventanas} = \left\lfloor \frac{\text{Longitud de la señal} - \text{Longitud de ventana}}{\text{Paso}} \right\rfloor + 1 \quad (16)$$

Utilizando la ecuación 16 con respecto a la señal con mayor número de muestras, una tamaño de ventana de 400 y un paso de 10, obtenemos un total de ventanas igual a 25 para cada segmento. Donde utilizando la ecuación 15 y multiplicando dicho valor con el total de ventanas obtenemos un total de 62500 segmentos de señal.

Finalmente, la tabla 3.8, compara el incremento de datos posterior al procesamiento aplicado.

Por movimiento		
Datos no segmentados	Datos segmentados	Windowed data
500 señales	2500 señales	62500 segmentos

Cuadro 3.8: Total de señales con datos no segmentados, segmentados y con ventaneo.

Cada ventana creada a partir de los segmentos contiene la información correspondiente a los 8 sensores, ya que, como vimos en secciones anteriores, los tiempos correspondientes a la toma de datos es la misma para cada movimiento y sujeto.

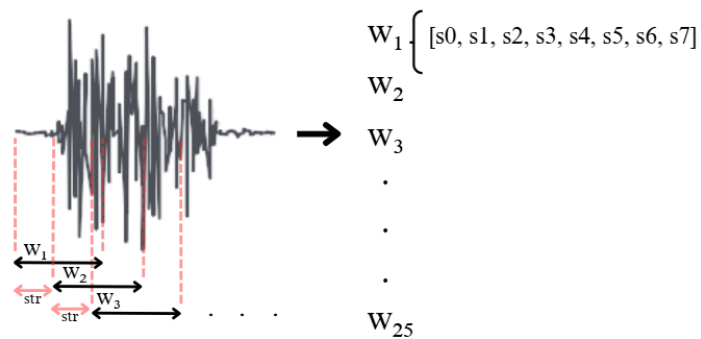


Figura 3.24: Agrupación de las muestras tras aplicar el ventaneo en cada segmento.

La figura 3.24, describe los datos pertenecientes a cada ventana. Donde W describe el número de ventana y S describe al número de sensor. La segmentación toma 400 datos totales de cada segmento. Sin embargo, se mantiene la lógica de independencia entre sensores. Teniendo finalmente datos de tipo (número de ventanas, longitud de las

ventanas, número de características).

Una vez que los datos han sido segmentados, se logra un aumento en la cantidad de datos mediante la técnica de ventaneo. Aunque no se están generando señales nuevas o artificiales, el proceso de ventaneo permite tratar cada ventana como una unidad de datos independiente, facilitando su análisis.

A continuación se presenta la clasificación de los movimientos utilizando los datos previamente procesados y segmentados utilizando diferentes metodologías para analizar el comportamiento con respecto a distintos enfoques de resolución.

3.6. Clasificación

Una vez obtenido el procesamiento necesario para nuestro objetivo, es importante analizar qué clasificador utilizar. Un clasificador debe ser capaz de identificar patrones distintivos en los datos obtenidos de las señales EMG y asignar correctamente cada patrón a una clase específica.

Como se ha discutido en secciones anteriores, cada clasificador tiene características específicas para realizar la extracción de características. En este caso, nuestro objetivo es realizar la extracción de características de forma automática. Las CNN son ideales para este tipo de tareas, ya que su efectividad en la extracción de características ha sido ampliamente demostrada [92, 93]. Sin embargo, uno de los principales desafíos que enfrentamos es el análisis de series de tiempo, dado que la señal EMG es no estacionaria, lo que dificulta su análisis automático. Este problema puede mitigarse limitando el análisis de la señal a ventanas de tiempo fijas, como se ha señalado en secciones previas. Además, existen metodologías avanzadas que son capaces de aprender patrones a lo largo del tiempo, como las redes LSTM y los modelos basados en transformers, las cuales ofrecen soluciones prometedoras para este tipo de datos.

En este trabajo, se han utilizado tres clasificadores diferentes: primero, un clasificador basado en CNN; segundo, una combinación de las características que nos ofrece un extractor de características como CNN y un clasificador LSTM, obteniendo un clasificador CNN-LSTM; y, finalmente, un clasificador basado en transformers denominado PatchTST. Es importante mencionar también los elementos de configuración con respecto a cada metodología, pues como se describió previamente, no todos los parámetros o elementos de configuración son los mismos para los clasificadores. A continuación se describen los hiperparámetros de nuestro interés.

- **Función de activación:** La función de activación permite cuantificar qué tan cerca de los resultados en base a los errores producidos en el entrenamiento. Su propósito es introducir no linealidad en el modelo.
- **Pesos:** Los pesos nos ayudan a determinar la fuerza de las conexiones entre las neuronas del modelo. Los pesos se actualizan con respecto al entrenamiento.

- **Optimizador:** El optimizador permite ajustar los pesos de la red neuronal durante el proceso de entrenamiento para minimizar la función de pérdida.
- **Función de pérdida:** Es una medida que cuantifica qué tan bien o mal está funcionando el modelo en relación con los resultados esperados. Para ello, se calcula la diferencia entre las predicciones del modelo y las etiquetas reales. Durante el entrenamiento, el objetivo es minimizar esta función de pérdida, ajustando los pesos de la red para mejorar la precisión de las predicciones.
- **Tamaño del lote:** Considerando cada ventana de la señal. Es el número de ejemplos que se procesan en cada paso de actualización de los pesos durante el entrenamiento.
- **Épocas:** Una época es un ciclo completo en el que el modelo pasa por todo el conjunto de datos de entrenamiento una vez. El número de épocas determina cuántas veces el modelo ve todos los ejemplos en el entrenamiento.
- **Early stopping:** Es una técnica que permite monitorear el rendimiento del modelo en el conjunto de validación durante el entrenamiento en base a los resultados que se obtienen. Si el rendimiento en validación deja de mejorar después de un número determinado de épocas, el entrenamiento se detiene, y los pesos del modelo se restauran a los mejores valores alcanzados.
- **Unidades densas:** Se refiere al número de neuronas que se encuentran totalmente conectadas a todas las neuronas de la capa anterior.
- **Tasa de aprendizaje:** La tasa de aprendizaje determina el tamaño de los ajustes que se hacen a los pesos del modelo en cada iteración del entrenamiento.
- **Kernel:** Es una pequeña matriz de valores que se aplica a los datos de entrada para realizar la operación de convolución.
- **Tamaño del kernel:** Este hiperparámetro es específico de una CNN, el kernel es el tamaño del filtro que se desliza sobre los datos de entrada para extraer características.
- **Número de capas:** Como su nombre lo indica, se refiere a la cantidad de capas en una red neuronal, lo que determina su profundidad.
- **Unidades recurrentes:** Las unidades recurrentes son específicas de las RNN o derivados (en nuestro caso de la red LSTM), y nos permiten que la red mantenga información de pasos anteriores y las use para predecir la salida en pasos futuros.
- **Tamaño del parche:** Este hiperparámetro es específico del modelo PatchTST, nos determina cuántos "parches" estarán involucrados con respecto a la división de los datos de entrada, para ser procesados independientemente.

- **Cabezas:** Este hiperparámetro se aplica al modelo PatchTST, las "cabezas" se refieren a las múltiples atenciones paralelas que el modelo realiza. Cada cabeza de atención tiene su propio conjunto de parámetros y aprende diferentes aspectos de las relaciones entre las entradas.

Cada hiperparámetro es ajustable, y dicho ajuste depende de la respuesta observada durante el entrenamiento.

Dentro de los entrenamientos se utilizaron diversas combinaciones las cuales fueron modificadas a lo largo de los resultados. Sin embargo, los mejores resultados fueron obtenidos mediante los hiperparámetros descritos en la tabla 3.9

Función de activación	Tamaño del kernel	Filtros	Optimizador	Función de pérdida	Tamaño de lote	Épocas
ReLU/Softmax	3	64	Adam	Entropía cruzada categórica dispersa	32	50

Cuadro 3.9: Primeros hiperparámetros utilizados

Posteriormente se aplicó una metodología basada en la búsqueda de la mejor combinación de hiperparámetros con respecto al resultado en los entrenamientos, para ello se utilizó una optimización con dos técnicas: Grid Search y Gaussian Process.

- **Búsqueda en cuadrícula (GS):** Es una técnica de búsqueda que evalúa un conjunto predefinido de valores para cada hiperparámetro. En este enfoque, se define una "malla" con diferentes combinaciones de hiperparámetros y se evalúan todas las combinaciones posibles. [94]
- **Proceso Gaussiano (GP):** Esta técnica permite la predicción de combinaciones que otorgarán el mejor rendimiento en el entrenamiento. A diferencia de la búsqueda grid search, al utilizar este método no existe la necesidad de evaluar todas las opciones posibles, ya que se utiliza un proceso iterativo que ajusta la búsqueda en función de los resultados anteriores, enfocándose en áreas del espacio de hiperparámetros que tienen más potencial para mejorar el rendimiento del modelo. [95]

Posteriormente a la búsqueda de hiperparámetros con GS y GP, se obtuvo que, la función de activación utilizada para los tres casos es ReLU. Los pesos en todos los modelos se inicializan aleatoriamente y se ajustan mediante el optimizador AdamW, conocido por su eficiencia computacional y estabilidad durante el entrenamiento. La función de pérdida utilizada en todos los modelos es categorical cross-entropy, la cual es ampliamente usada para tareas de clasificación multiclase. Además, se realizó a conversión de las etiquetas a formato one-hot encoding, lo que significa que cada

etiqueta se representa como un vector binario en el que solo una posición tiene el valor 1 (indicando la clase verdadera), mientras que las demás posiciones tienen el valor 0. La tabla 3.10, describe la conversión de las etiquetas reales al formato one-hot encoding.

Movimiento	Etiqueta real	Etiquetas con codificación one-hot
Desviación cubital	0	0000000001
Flexión	1	0000000010
Pinza gruesa	2	0000000100
Posición inicial	3	0000001000
Extensión	4	0000010000
Pronación	5	0000100000
Supinación	6	0001000000
Pinza fina	7	0010000000
Desviación radial	8	0100000000
Expansión	9	1000000000

Cuadro 3.10: Comparación entre etiquetas

El objetivo del entrenamiento es minimizar la pérdida obtenida, mediante el ajuste de los pesos que inicialmente son aleatorios, para que las probabilidades predichas se acerquen lo más posible al vector one-hot de las etiquetas reales.

El tamaño del lote varía entre los modelos: 32 para CNN y CNN-LSTM, y 64 para PatchTST, debido a las diferencias en la arquitectura y la naturaleza de los modelos. El número de épocas también difiere: 30 para los primeros dos modelos y 150 para PatchTST, lo que permite una convergencia más robusta en este último. Se implementó early stopping para evitar sobreajuste, configurado en 5 épocas sin mejora para CNN y CNN-LSTM, y 10 épocas para PatchTST. Las unidades densas son de 64 y 128 en CNN y CNN-LSTM, este hiperparámetro no aplica para PatchTST. La tasa de aprendizaje inicial es de 0.0001 en todos los casos, seleccionada para asegurar una convergencia estable. En cuanto al kernel, CNN y CNN-LSTM emplean tamaños de 3x3 y 5x5, mientras que PatchTST no lo utilizan. Las unidades recurrentes están presentes únicamente en CNN-LSTM, con configuraciones de 64 y 128. El modelo PatchTST introduce hiperparámetros específicos, como un tamaño de patch de 400 y 4 cabezas de atención. Finalmente, todos los modelos manejan 10 clases de salida, una entrada de dimensión 8 proporcionada por el número de sensores utilizados.

También se emplearon técnicas de regularización, como el dropout para prevenir el sobreajuste. El valor de dropout fue ajustado en 0.3.

4. Resultados y discusión

Para verificar que los resultados sean correctos, se utilizan métricas de evaluación del rendimiento. Estas métricas permiten analizar el comportamiento del clasificador durante el entrenamiento y evaluar la calidad de la clasificación obtenida. [96]

4.1. Evaluación de métricas

4.1.1. Matriz de confusión

La matriz de confusión se representa como una tabla que compara las predicciones del clasificador con los valores reales de las clases, proporcionando un desglose detallado de los errores y aciertos en la clasificación. La figura 4.25, muestra la representación de una matriz de confusión.

	Positivo	FP
Valores reales	TP	FN
Negativo	FN	TN
	Positivo	Negativo
	Valores predichos	

Figura 4.25: Representación de una matriz de confusión. [97]

Cuando uno de los datos verdaderos es clasificado de forma correcta se considera un verdadero positivo (TP), cuando uno de los datos negativos se clasifica correctamente se considera un verdadero negativo (TN). Por otro lado, cuando un dato verdadero es clasificado como negativo, se considera un falso negativo (FN) y, cuando un dato negativo es clasificado como verdadero, se considera un falso positivo (FP). Es en base a estos valores que se pueden describir las métricas para la evaluación del clasificador. [98]

4.1.2. Accuracy

Accuracy es una de las métricas más importantes en la clasificación, ya que mide la proporción de datos clasificados correctamente con respecto al total de datos clasificados. La ecuación (17) representa el cálculo del accuracy del clasificador.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

4.1.3. Precisión

La precisión indica cuántas de las predicciones positivas hechas por el clasificador son realmente correctas. Para calcularla, se utilizan todos los datos clasificados como positivos y se dividen entre el total de predicciones positivas realizadas. La ecuación (18) describe el cálculo matemático para la precisión.

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (18)$$

4.1.4. Recall

La sensibilidad o Recall mide aquellos valores clasificados como positivos dentro del conjunto de datos. Es decir, calcula cuántos de los casos positivos reales fueron correctamente identificados como positivos, considerando también aquellos casos positivos que fueron incorrectamente clasificados como negativos. La ecuación (19) calcula el Recall del clasificador.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (19)$$

4.1.5. Especificidad

La especificidad evalúa la capacidad del clasificador para identificar correctamente las instancias negativas, es decir, cuántos de los casos negativos reales fueron correctamente clasificados como negativos, y tiene en cuenta los falsos positivos, es decir, aquellos casos negativos que fueron incorrectamente clasificados como positivos. La ecuación (20) describe los valores a utilizar para generar la métrica.

$$\text{Especificidad} = \frac{TN}{TN + FP} \quad (20)$$

4.1.6. F1-Score

El F1 score es una métrica que combina la precisión como la sensibilidad en un solo valor, proporcionando un balance entre estos dos aspectos. Es especialmente útil cuando hay un desbalance entre las clases en los datos, ya que tiene en cuenta tanto los falsos positivos como los falsos negativos. La ecuación (21) describe cómo calcular el F1-Score.

$$\text{F1-Score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}} \quad (21)$$

La ecuación (22) también describe en términos de las instancias calculadas en el clasificador el F1-Score.

$$\text{F1-Score} = 2 \times \frac{TP}{2 \times TP + FP + FN} \quad (22)$$

4.1.7. Curva ROC y AUC

La curva característica de operación del receptor (ROC) constituye un método estadístico para determinar la exactitud de un modelo, además de determinar el punto de corte en el que se alcanza la sensibilidad y especificidad más alta. [99] La curva ROC se representa en un gráfico de dos dimensiones, donde el eje X representa la tasa de falsos positivos (FPR) y el eje Y la tasa de verdaderos positivos (TPR). Tal como se representa en la figura 4.26.

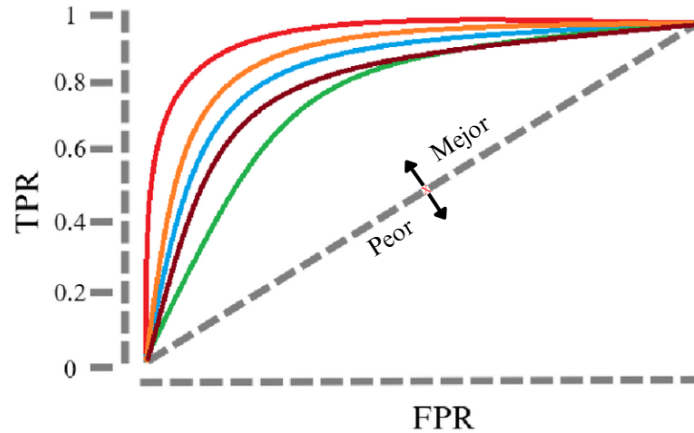


Figura 4.26: Representación de una curva ROC. Adaptado de [100]

El objetivo de la gráfica es representar qué tan efectivo es un clasificador. Cada uno de los valores son graficados punto a punto hasta conseguir una curva, el objetivo es conseguir una curva cercana al valor 1. Los valores TPR y FPR son calculados con las ecuaciones (23) y (24).

$$\text{FPR} = \frac{FP}{FP + TN} \quad (23)$$

$$\text{TPR} = \frac{TP}{TP + FN} \quad (24)$$

El área bajo la curva (AUC), es el valor utilizado para calcular la tendencia que sigue la curva generada por cada uno de los puntos mediante la ecuación (25).

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \quad (25)$$

Un **AUC** = 1 indica un clasificador perfecto, capaz de distinguir completamente entre clases positivas y negativas. Un **AUC** = 0.5 representa un clasificador sin capacidad discriminativa, equivalente a una clasificación aleatoria. Un **AUC** < 0.5 sugiere un modelo con un desempeño inferior a la aleatoriedad, lo que implica que invirtiendo su decisión se podría obtener una mejor clasificación.

4.1.8. Curvas de pérdida

La curva de pérdida es un gráfico que muestra cómo cambia la función de pérdida durante el entrenamiento del clasificador. El objetivo del gráfico es representar qué tan cerca se encuentra la pérdida del valor cero. A lo largo de las épocas del entrenamiento, la curva de pérdida evalúa la diferencia entre las predicciones del modelo y los valores reales del conjunto de datos. La figura 4.27, representa la evolución de la función de pérdida a lo largo de las épocas de entrenamiento.

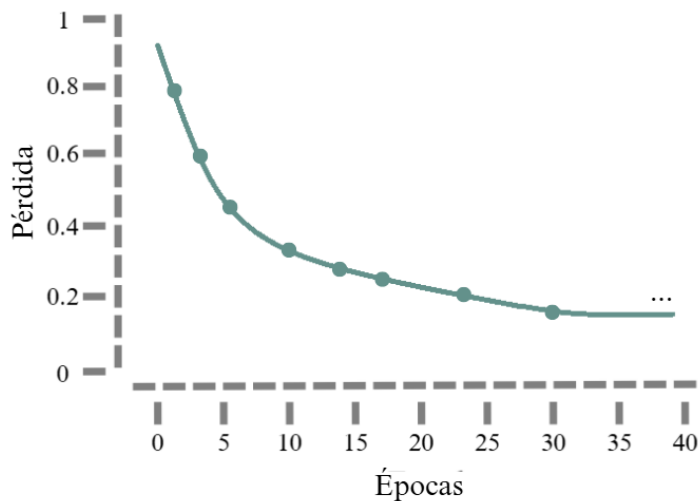


Figura 4.27: Representación de una curva de pérdida. Adaptado de [101]

4.1.9. t-SNE

En algunas metodologías, es posible visualizar características de alta dimensionalidad en un espacio de menor dimensionalidad aplicando incrustación estocástica de vecinos distribuida en t (t-SNE), una técnica que permite comprender la semejanza entre características de cada grupo.

Para su cálculo, t-SNE considera la probabilidad de que dos puntos sean vecinos, basándose en la proximidad de los puntos en el espacio de alta dimensión. Esta probabilidad se mide utilizando una distribución Gaussiana alrededor de cada punto. Así, los puntos más cercanos tienen una probabilidad más alta de ser vecinos, reflejando una relación fuerte entre ellos, mientras que los puntos más distantes tienen una probabilidad baja, lo que indica una relación débil. Posteriormente, t-SNE reduce esta información a un espacio de baja dimensión, manteniendo las relaciones de proximidad entre los puntos. [102]

A pesar de que cada clasificador proporciona características distintas, t-SNE no suele ser ideal cuando las características extraídas tienen una alta dimensionalidad y estructura compleja, como en el caso de modelos basados en Transformers, específicamente en nuestro caso con PatchTST. Esto se debe a que t-SNE tiende a perder información

estructural cuando la dimensionalidad de los datos originales es muy alta, lo que puede afectar la interpretación de los resultados en el espacio reducido.

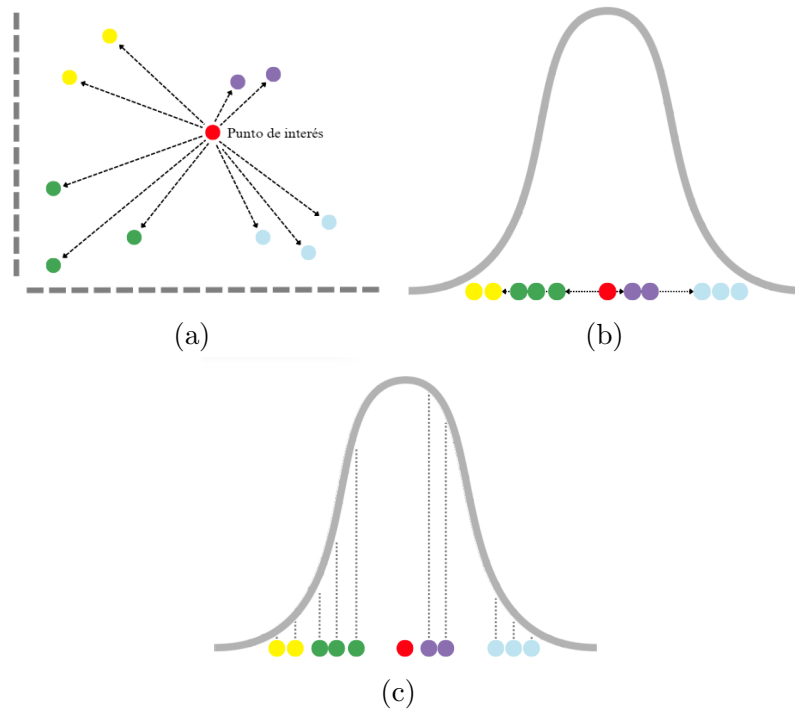


Figura 4.28: Proceso de reducción de la dimensionalidad mediante t-SNE: (a) Medición de las distancias entre el punto de interés y los demás puntos, (b) Muestreo de las distancias en una distribución gaussiana, (c) Cálculo de la puntuación de similitud entre los puntos. Adaptado de [102].

La figura 4.28 ilustra cómo los puntos se agrupan en el espacio, donde los puntos más cercanos en el espacio original tienden a mantenerse cercanos en la representación de baja dimensión, lo que facilita la interpretación de las relaciones y agrupaciones en los datos.

En la figura 4.28(a), se calcula la distancia euclidiana entre los puntos en el espacio de alta dimensión. Posteriormente, estas distancias se utilizan para definir una distribución de probabilidades basada en una distribución Gaussiana, como se muestra en la figura 4.28(b). Finalmente, en la figura 4.28(c), se calculan las probabilidades de semejanza con respecto a un punto de interés. El valor máximo de la distribución corresponde al propio punto (probabilidad = 1). Los puntos más cercanos al centro de esta distribución tienen una mayor probabilidad de ser considerados vecinos del punto de interés. El objetivo es que las similitudes en el espacio original se conserven en el espacio reducido, permitiendo identificar agrupaciones o estructuras en los datos.

4.1.10. P-valor

El P-valor es una metodología estadística que permite evaluar una hipótesis a través de una probabilidad. Es decir, permite determinar si la diferencia observada entre dos grupos es estadísticamente significativa o si podría haber ocurrido por azar.

Para ello, se utilizan dos casos o hipótesis: hipótesis nula (H_0), la cual nos determina que no hay una diferencia real entre las dos observaciones e hipótesis alternativa (H_1), la cual nos determina que sí hay una diferencia real entre las dos observaciones. Esto basado en un umbral de 0.05, es decir, si el p-valor obtenido para el análisis de las dos muestras es menor o igual a 0.5, se considera que hay suficiente evidencia para rechazar H_0 en favor de H_1 . Por otro lado, si el p-valor es mayor a 0.05, no se rechaza la hipótesis nula, lo que implica que no hay suficiente evidencia para afirmar que exista una diferencia real entre las observaciones. [103, 104]

Es importante destacar que las hipótesis H_0 y H_1 se definen en función de la problemática o el contexto del estudio en el que se esté trabajando. Sin embargo, la metodología para calcular el p-valor es la misma en todos los casos.

La elección de la prueba estadística a utilizar depende del tipo de estudio que se esté realizando.

Cada prueba estadística se aplica de manera diferente, lo que implica que los valores a calcular varían según la prueba utilizada. En nuestro caso de estudio, se empleó la prueba t-Student, pues nuestro objetivo es analizar las diferencias obtenidas entre dos muestras. [105]

Para ello, en cada par de observaciones se calculan las diferencias mediante la ecuación (26)

$$d_i = X_{1i} - X_{2i} \quad (26)$$

Donde X_{1i} es el valor de la primera muestra en la agrupación i , y X_{2i} es el valor de la segunda muestra en la agrupación i .

Posteriormente se calcula la media obtenida para el total de las diferencias, de esta forma se obtiene el valor promedio entre las dos muestras. Para ello, se utiliza la ecuación (27)

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (27)$$

Una vez obtenida la media de las diferencias, se procede a calcular la desviación estándar de las diferencias utilizando la ecuación (28). De esta forma se analiza la dispersión de las diferencias con respecto a la media, además de ser necesaria para el cálculo de la prueba t-Student.

$$s_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (28)$$

Finalmente, es posible calcular el valor de la estadística t-Student con los valores obtenidos anteriormente (\bar{d} y s_d) y la ecuación (29)

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}} \quad (29)$$

Bajo este enfoque, entonces, la prueba t-Student nos permite comparar las métricas de rendimiento, dado que los modelos fueron entrenados y evaluados bajo condiciones similares.

4.2. Resultados

Las métricas nos permiten analizar y comprender el comportamiento de un clasificador. Sin embargo, también constituyen el punto de partida para evaluar si nuestros datos se encuentran en un estado óptimo para aplicarles una clasificación.

Como se describió en secciones anteriores, cada clasificador posee características e hiperparámetros distintos. Sin embargo, es posible aplicar las métricas de evaluación para obtener un análisis más efectivo con respecto a los resultados. A continuación se describen las métricas obtenidas para cada caso.

Es importante de primera instancia analizar el rendimiento obtenido en el entrenamiento del modelo, pues apesar de concluir efectivamente y obtener las demás métricas es necesario visualizar la evolución de nuestro algoritmo a lo largo de las épocas.

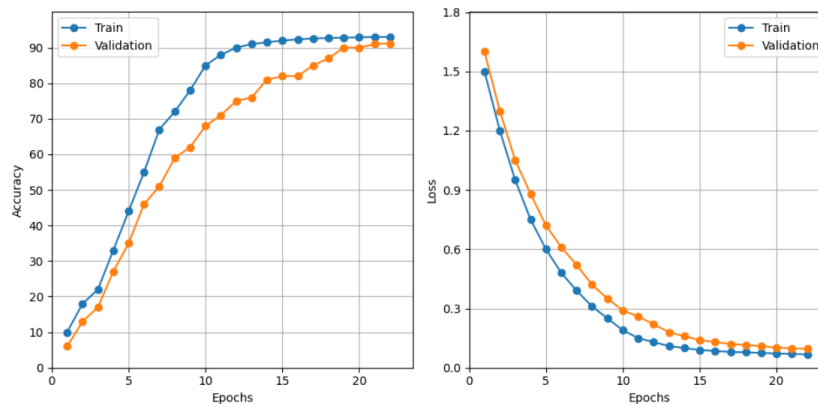


Figura 4.29: Curvas de rendimiento del clasificador CNN.

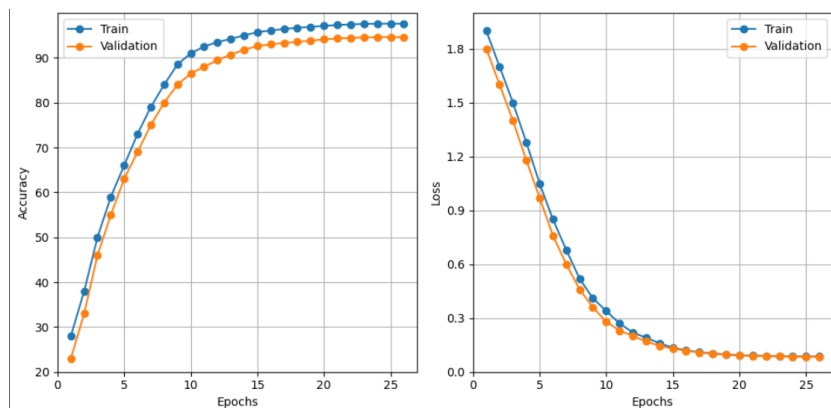


Figura 4.30: Curvas de rendimiento del clasificador CNN-LSTM.

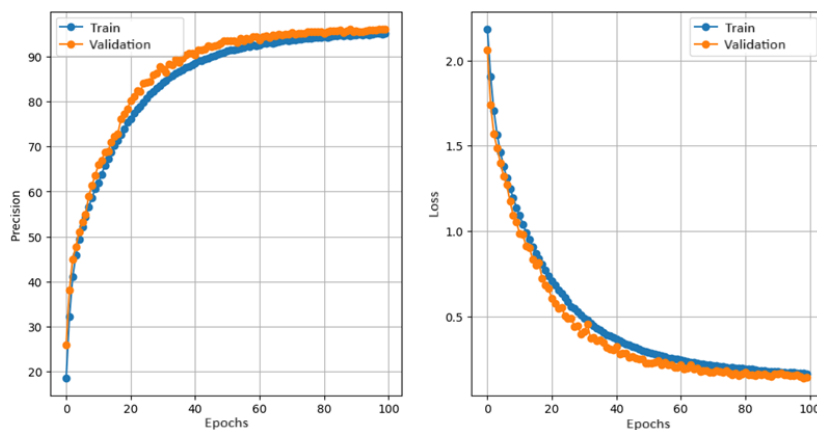


Figura 4.31: Curvas de rendimiento del clasificador PatchTST.

Se presentan dos gráficos de acuerdo a las figuras 4.29, 4.30 y 4.31, el primer gráfico describe la evolución del accuracy obtenido a lo largo de las épocas y el segundo gráfico describe la curva de pérdida de nuestro mejor resultado obtenido de todos los entrenamientos realizados para cada clasificador. En los gráficos, se aprecia la evolución a lo largo de las épocas en el conjunto de entrenamiento y validacion. Para los dos primeros casos, es notable cómo apesar de tener una curva muy cercana en ambos conjuntos. por otro lado, para el tercer caso, la curva de accuracy tiende a ir muy similar en ambos conjuntos, esto principalmente por la cantidad de épocas de entrenamiento que se utilizan en para el clasificador PatchTST. Sin embargo, es evidente el ajuste en los 3 casos.

También se analiza la matriz de confusión, ya que, además de obtener una visión general de cómo se comporta el modelo con cada clase, también es el punto de partida para calcular nuestras métricas.

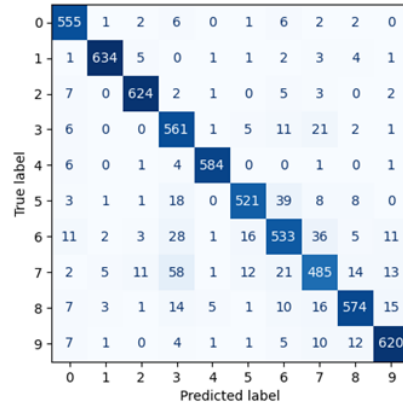


Figura 4.32: Matriz de confusión del modelo CNN

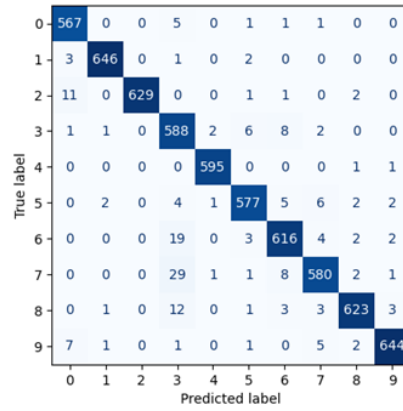


Figura 4.33: Matriz de confusión del modelo CNN-LSTM

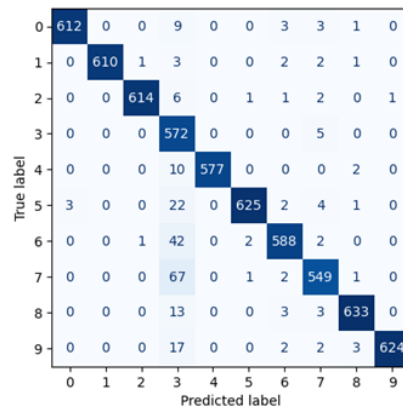


Figura 4.34: Matriz de confusión del modelo PatchTST

La Figura 4.32 presenta los resultados de clasificación para las 10 clases utilizando un modelo basado en CNN. De manera análoga, la Figura 4.33 muestra los resultados obtenidos mediante CNN-LSTM, mientras que la Figura 4.34 corresponde al clasificador PatchTST.

Además del análisis cuantitativo de las predicciones, la representación gráfica de los resultados permite identificar patrones en el desempeño de cada modelo a través de la intensidad del color en las celdas de la matriz de confusión.

Si bien cada clasificador presenta diferencias en su desempeño, se observa una dificultad recurrente en la clasificación de la clase 4 (etiquetada como 3), lo que sugiere una alta complejidad en las características asociadas a esta categoría. Asimismo, se evidencia un patrón de falsos positivos (FP) con las clases 7, 8 y 9, indicando posibles similitudes en la distribución de sus características. En particular, el modelo basado en CNN presenta la mayor tasa de falsos positivos, lo que sugiere una menor capacidad de discriminación en comparación con los otros dos clasificadores.

Una de las dificultades al utilizar la matriz de confusión como métrica de análisis es que no es muy viable para el análisis de inferencia del clasificador. Por ende se presentan un análisis cuantitativo basado en las métricas accuracy, precision, recall y f1-score.

El accuracy, basado en el estado del arte, es la principal métrica de análisis para los clasificadores. La tabla 4.11 representa los tres clasificadores y el accuracy obtenido en el mejor entrenamiento de cada modelo.

	CNN	CNN-LSTM	PatchTST
Accuracy	91.06 %	97.04 %	96.06 %

Cuadro 4.11: Comparación del accuracy obtenido para cada modelo.

Cada clasificador nos otorgó métricas superiores al 90 %, lo que es buen indicador si se trata de un clasificador. La tabla nos muestra también la predominancia del clasificador basado en CNN-LSTM, pues al combinar una arquitectura basada en las redes convolucionales como extractor de características y un modelo basado en las redes recurrentes, es posible aprender e interpretar mejor las características de cada muestra. Por otro lado, el clasificador basado en transformers también resultó eficiente para la tarea de clasificación. Sin embargo, es importante considerar que este tipo de arquitectura requiere de una gran cantidad de datos para alcanzar resultados más completos.

Como complemento al accuracy se presentan las tablas 4.12, 4.13 y 4.14 otras métricas para el análisis de cada clasificador.

Tal como el caso de las matrices de confusión, cada clasificador nos proporcionó resultados diferentes. Sin embargo, notamos una ligera inclinación al clasificador CNN-LSTM con respecto al clasificador PatchTST y el clasificador CNN. Y de igual forma,

CNN			
Movimiento	Precisión	Recall	F1-score
0	91.74 %	96.52 %	94.07 %
1	97.99 %	97.24 %	97.61 %
2	96.30 %	96.89 %	96.59 %
3	80.72 %	92.27 %	86.11 %
4	98.15 %	97.82 %	97.99 %
5	93.37 %	86.98 %	90.06 %
6	84.34 %	82.51 %	83.41 %
7	82.91 %	77.97 %	80.36 %
8	92.43 %	88.85 %	90.61 %
9	93.37 %	93.80 %	93.58 %

Cuadro 4.12: Métricas de rendimiento para el modelo CNN.

CNN-LSTM			
Movimiento	Precisión	Recall	F1-score
0	96.26 %	98.61 %	97.42 %
1	99.23 %	99.08 %	99.16 %
2	1 %	97.67 %	98.82 %
3	89.23 %	96.71 %	92.82 %
4	99.33 %	99.66 %	99.50 %
5	97.30 %	96.33 %	96.81 %
6	95.95 %	95.36 %	95.65 %
7	96.51 %	93.25 %	94.85 %
8	98.26 %	96.44 %	97.34 %
9	98.62 %	97.43 %	98.02 %

Cuadro 4.13: Métricas de rendimiento para el modelo CNN-LSTM.

PatchTST			
Movimiento	Precisión	Recall	F1-score
0	99.51 %	97.45 %	98.47 %
1	1 %	98.55 %	99.27 %
2	99.68 %	98.24 %	98.95 %
3	84.91 %	99.27 %	86.72 %
4	1 %	97.96 %	98.97 %
5	99.36 %	95.13 %	97.20 %
6	97.51 %	92.60 %	94.99 %
7	95.98 %	88.55 %	92.11 %
8	98.60 %	97.09 %	97.84 %
9	99.84 %	96.30 %	98.04 %

Cuadro 4.14: Métricas de rendimiento para el modelo PatchTST.

notamos un desempeño menos eficiente en la clase 4.

En las figuras 4.35, 4.36 y 4.37 se representa visualmente cómo se distribuyen las 4 métricas previamente descritas, donde la barra de color azul describe el accuracy, la verde describe la precisión, la amarilla el recall y la roja el f1-score.

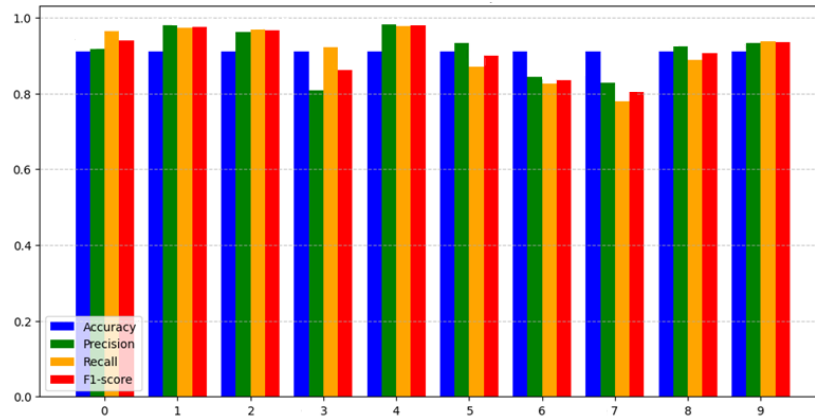


Figura 4.35: Gráfico de barras de las métricas de rendimiento del modelo CNN.

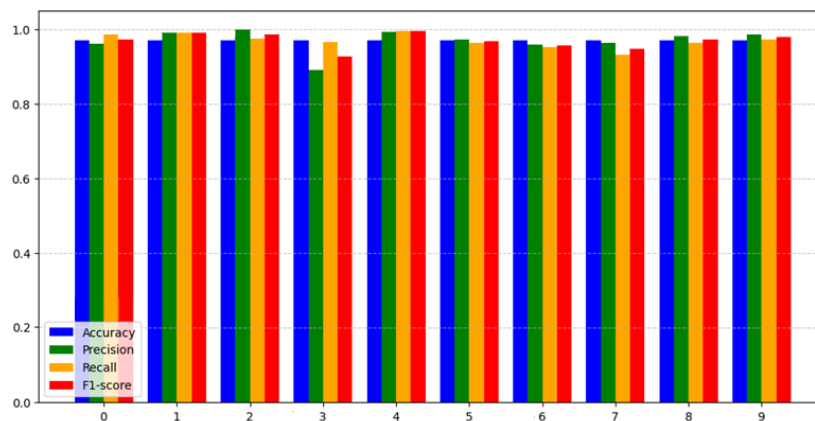


Figura 4.36: Gráfico de barras de las métricas de rendimiento del modelo CNN-LSTM.

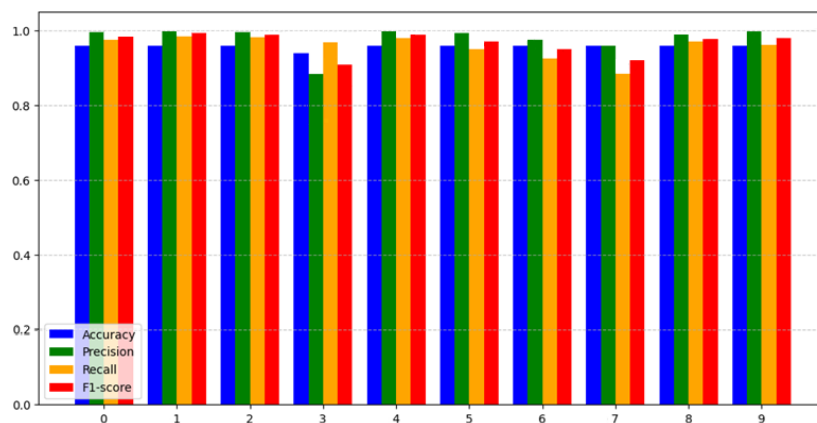


Figura 4.37: Gráfico de barras de las métricas de rendimiento del modelo PatchTST.

Las gráficas de barra nos presentan una representación más efectiva sobre el comportamiento de cada clasificador. Es evidente un rendimiento no tan efectivo en la clase 4, siendo esta la clase que nos disminuye las métricas de rendimiento, pues al analizar las demás clases, la clasificación se presentó de forma eficiente, sobre todo en el clasificador basado en CNN-LSTM.

Como parte del análisis de la exactitud de los clasificadores se presentan las gráficas 4.38, 4.39 y 4.40, las cuales nos describen la curva ROC de cada modelo, donde cada color describe una clase.

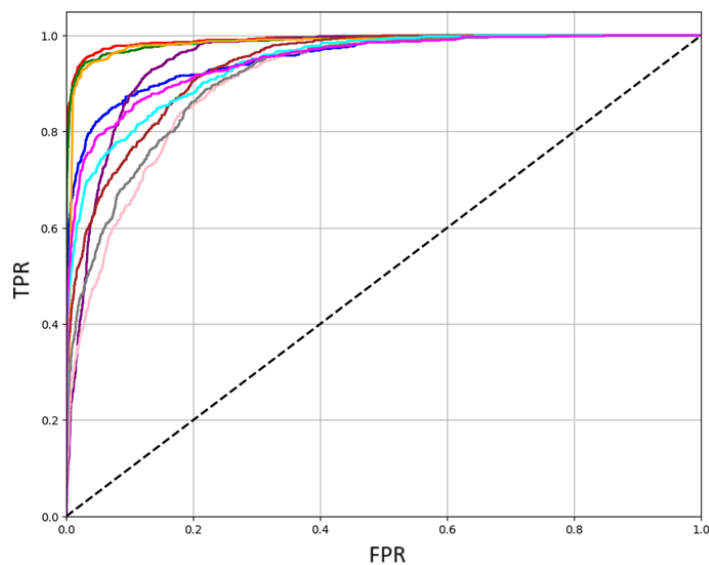


Figura 4.38: Curva ROC del modelo CNN.

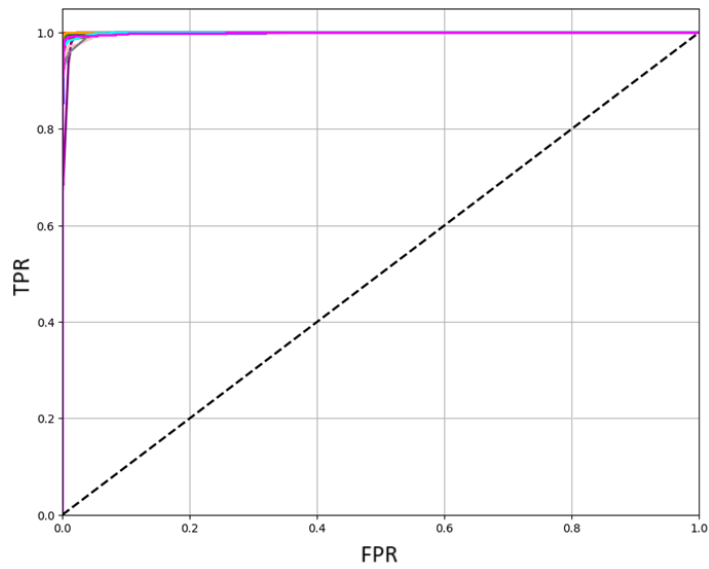


Figura 4.39: Curva ROC del modelo CNN-LSTM.

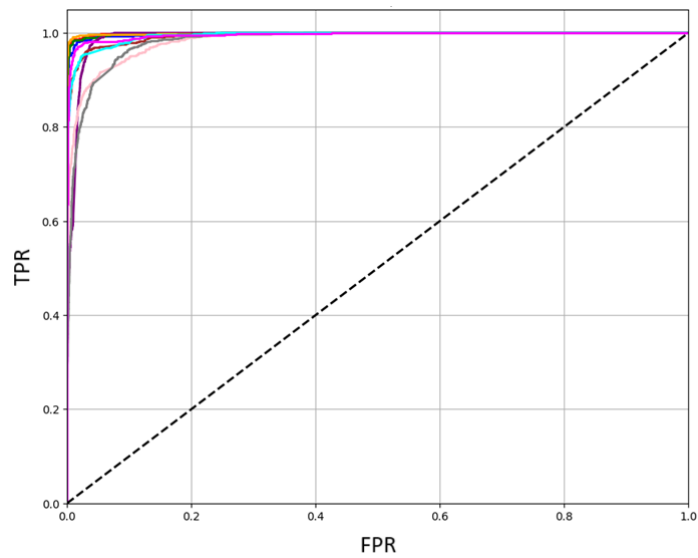


Figura 4.40: Curva ROC del modelo PatchTST.

Estos gráficos nos demuestran la efectividad del modelo CNN-LSTM para la clasificación de los movimientos, al igual que el modelo PatchTST que apesar de no tener una cantidad ideal de datos para el entrenamiento, nos otorgó resultados significativos. Para un análisis más detallado, las tablas 4.15, 4.16 y 4.17 presentan los valores de AUC de cada clase para cada modelo

CNN		
Movimiento	Color	Valor AUC
0	Azul	0.82
1	Rojo	0.94
2	Verde	0.93
3	Morado	0.91
4	Amarillo	0.932
5	Café	0.81
6	Rosa	0.79
7	Gris	0.797
8	Celeste	0.926
9	Lila	0.83
Media		0.8685

Cuadro 4.15: Valores AUC para el clasificador CNN.

CNN-LSTM		
Movimiento	Color	Valor AUC
0	Azul	0.97
1	Rojo	0.98
2	Verde	0.98
3	Morado	0.984
4	Amarillo	0.982
5	Café	0.97
6	Rosa	0.99
7	Gris	0.987
8	Sky blue	0.99
9	Lila	0.98
Media		0.9813

Cuadro 4.16: Valores AUC para el clasificador CNN-LSTM.

PatchTST classifier		
Movimiento	Color	Valor AUC
0	Azul	0.95
1	Rojo	0.97
2	Verde	0.98
3	Morado	0.95
4	Amarillo	0.98
5	Café	0.95
6	Rosa	0.93
7	Gris	0.94
8	Celeste	0.96
9	Lila	0.94
Media		0.9519

Cuadro 4.17: Valores AUC para el clasificador PatchTST.

De acuerdo a las tablas, se presenta una similitud entre los resultados obtenidos para el clasificador CNN y el clasificador PatchTST. Donde también se visualiza que el clasificador CNN presenta determinadas dificultades al identificar los patrones de las muestras.

También, se presenta la visualización t-SNE para los modelos CNN y CNN-LSTM. Donde es posible identificar las agrupaciones generadas para cada clase. Sin embargo, su aplicación en modelos basados en Transformers, en nuestro caso, PatchTST, presenta limitaciones significativas debido a la naturaleza de las representaciones generadas por

el clasificador.

Las figuras 4.41 y 4.42 representan el t-SNE de CNN y CNN-LSTM respectivamente.

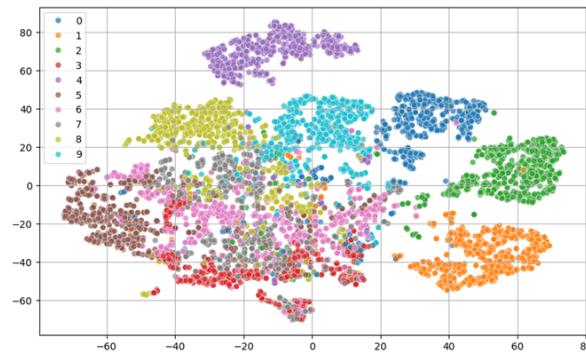


Figura 4.41: t-SNE para el modelo CNN.

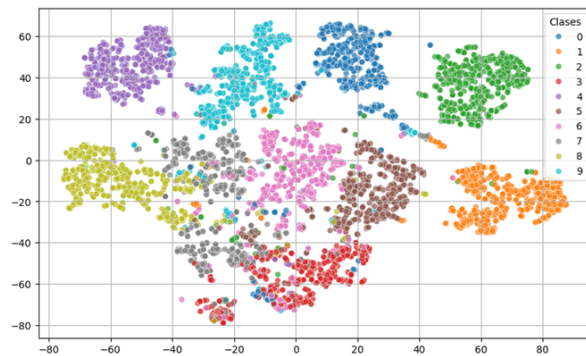


Figura 4.42: t-SNE para el modelo CNN-LSTM.

En la agrupación por t-SNE es posible identificar las características para cada clase de forma aislada una de la otra. A pesar de que el modelo CNN presenta un aislamiento bueno, no es del todo efectivo. Por otro lado, el modelo CNN-LSTM nos presenta un aislamiento de clases más completo. Sin embargo, con cierto margen de mejora. Además, se presenta la tabla 4.18, que compara los resultados obtenidos con respecto a los trabajos presentados en el estado del arte, obteniendo mejora en los 3 modelos presentados.

Too, J. et al. [9]	Huang, D. & Chen, B. [10]	Bakircioğlu, K. & Ozkurt, N. [11]	Núñez, B. et al. [12]	Este trabajo
86.18 %	79.33 %	95.90 %	86.78 %	91.06 %
88.04 %				97.04 %
				96.06 %

Cuadro 4.18: Comparación de los accuracy obtenidos en el estado del arte y este trabajo.

Para finalizar el análisis de los resultados se aplica la metodología para el cálculo del P-valor.

En nuestro caso de interés, se plantearon las siguientes hipótesis:

- H_0 = No hay diferencia significativa en el rendimiento entre los tres clasificadores.
- H_1 = Al menos uno de los modelos tiene un rendimiento significativamente diferente de los otros.

Bajo este enfoque, se busca comprobar si el rendimiento del clasificador CNN-LSTM es significativamente diferente al rendimiento de los otros clasificadores. Es decir, de manera estadística, se evalúa si el clasificador CNN-LSTM muestra un rendimiento superior a lo que se esperaría en un escenario aleatorio.

Utilizando las ecuaciones (26), (27), (28) y (29), se calculó el valor del P-valor utilizando los valores de las tablas 4.15, 4.16 y 4.17. Donde, si el valor del P-valor obtenido es menor que el umbral de 0.05, se rechaza H_0 y se concluye que al menos uno de los clasificadores tiene un rendimiento significativamente diferente en comparación con los otros.

A continuación se presenta la tabla 4.19 con los resultados obtenidos:

	t-Student	P-valor
CNN vs CNN-LSTM	-5.6689	0.0003
CNN vs PatchTST	-5.6173	0.0003
CNN-LSTM vs PatchTST	3.5064	0.0067

Cuadro 4.19: Comparación del P-valor obtenido para los tres modelos.

En los tres casos comparativos, los P-valores son inferiores a 0.05, lo que indica que las diferencias entre los AUC de los clasificadores son estadísticamente significativas. Por lo tanto, H_0 debe ser descartada y en su lugar H_1 confirmada, indicando que existen diferencias estadísticamente significativas entre los clasificadores. Entonces, puede concluirse que, la diferencia entre el clasificador CNN y el clasificador CNN-LSTM es estadísticamente significativa, lo que indica que CNN-LSTM ofrece un mejor rendimiento que CNN según el valor AUC. También se encontró una diferencia significativa entre CNN y PatchTST, lo que sugiere que PatchTST supera a CNN en cuanto a rendimiento. Por último, analizando el P-valor y el valor medio de cada AUC obtenido por los clasificadores CNN-LSTM y PatchTST se demuestra que aunque ambos clasificadores son efectivos, CNN-LSTM tiene un rendimiento ligeramente superior en comparación con PatchTST.

5. Conclusiones

Trabajar con señales en series de tiempo, especialmente en el contexto de procesamiento y clasificación, es una tarea compleja, principalmente debido al alto costo computacional en el entrenamiento de los algoritmos. La contribución de este trabajo se centra en el procesamiento, segmentación y clasificación de señales utilizando algoritmos de aprendizaje profundo. Además, se propone el uso de modelos recientes basados en transformer, lo que nos permite adaptarnos a las nuevas tecnologías y aprovechar el rendimiento que estas ofrecen.

Los resultados del estudio revelaron que un procesamiento basado en el análisis de la señal en periodos cortos de tiempo facilita una correcta limpieza del ruido, que es una de las principales dificultades al trabajar con señales EMG. Por otro lado, la segmentación, implementada como parte de un aumento de datos en la metodología PatchTST, demostró ser funcional y eficiente en la clasificación de las señales EMG.

Es importante señalar que las metodologías de extracción automática de características tienen un amplio margen de mejora. Aunque actualmente son funcionales, las metodologías disponibles son relativamente simples. En este contexto, combinar modelos de aprendizaje profundo para extraer y analizar las características resulta ser un enfoque productivo, especialmente cuando se considera la eficiencia en términos de ahorro de tiempo.

Las señales bioeléctricas pueden ser problemáticas si no se comprende bien su naturaleza. Sin embargo, es más sencillo trabajar con ellas cuando se dispone de herramientas adecuadas para su análisis. Los trabajos previos han demostrado la efectividad de los algoritmos de aprendizaje profundo en la clasificación de señales, especialmente cuando se utilizan bases de datos dedicadas a esta tarea. Por ello, este trabajo propone un análisis de principio a fin para comprender las señales EMG en bruto, comparando su desempeño frente a situaciones no ideales.

Las señales bioeléctricas son complejas y altamente susceptibles a interferencias y variabilidad. A pesar de estas dificultades, los avances tecnológicos han permitido una comprensión más profunda de su naturaleza. El uso de herramientas especializadas para analizar señales de alta dimensionalidad es esencial para abordar estos desafíos. Los resultados obtenidos en este estudio indican que, al combinar un conocimiento detallado de la señal con técnicas de procesamiento adecuadas, es posible mitigar significativamente los problemas asociados con las señales EMG, mejorando su calidad y precisión para aplicaciones prácticas.

Referencias bibliográficas

- [1] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic myoelectric control strategies: A clinical perspective,” *Current Surgery Reports*, vol. 2, pp. 1–11, 2014.
- [2] C. Fang, B. He, Y. Wang, J. Cao, and S. Gao, “Emg-centered multisensory based technologies for pattern recognition in rehabilitation: State of the art and challenges,” *Biosensors*, vol. 10, no. 8, 2020.
- [3] 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. 12 431–12 466, 2013.
- [4] B. Peerdeman, D. Boere, H. Witteveen, R. Veld, H. Hermens, S. Stramigioli, J. Rietman, P. Veltink, and S. Misra, “Myoelectric forearm prostheses: State of the art from a user-centered perspective,” *Journal of rehabilitation research and development*, vol. 48, pp. 719–37, 2011.
- [5] J. B. M. A. E. M. Michael Sachs, M.D., “Historical evolution of limb amputation,” *World journal of surgery*, vol. 23, pp. 1088–93, 1999.
- [6] DGCS-UNAM, “75 amputaciones diarias en México,” *UNAM Global Revista*, 2022.
- [7] M. Asghari Oskoei and H. Hu, “Myoelectric control systems—a survey,” *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.
- [8] D. C. Toledo-Pérez, J. Rodríguez-Reséndiz, R. A. Gómez-Loenzo, and J. C. Jauregui-Correa, “Support vector machine-based emg signal classification techniques: A review,” *Applied Sciences*, vol. 9, no. 20, 2019.
- [9] J. Too, A. R. Abdullah, N. Mohd Saad, N. Mohd Ali, and T. N. S. Tengku Zawawi, “Featureless emg pattern recognition based on convolutional neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, pp. 1291–1297, 2019.
- [10] D. Huang and B. Chen, “Surface emg decoding for hand gestures based on spectrogram and cnn-lstm,” in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 2019, pp. 123–126.
- [11] K. Bakırcioğlu and N. Ozkurt, “Classification of emg signals using convolution neural network,” *International Journal of Applied Mathematics Electronics and Computers*, vol. 8, 2020.
- [12] B. Núñez Montoya, E. Valarezo Añazco, S. Guerrero, M. Valarezo-Añazco, D. Espin-Ramos, and C. Jiménez Farfán, “Myo transformer signal classification for an anthropomorphic robotic hand,” *Prosthesis*, vol. 5, 2023.

- [13] Y. N. Singh, S. K. Singh, and A. K. Ray, “Bioelectrical signals as emerging biometrics: Issues and challenges,” *International Scholarly Research Notices*, vol. 2012, no. 1, p. 712032, 2012.
- [14] R. M. Rangayyan, *Introduction to Biomedical Signals*. John Wiley Sons, Ltd, 2015, ch. 1, pp. 1–69.
- [15] L. Sörnmo and P. Laguna, “Chapter 1 - introduction,” in *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Academic Press, 2005, pp. 1–24.
- [16] A. Pal, A. K. Gautam, and Y. N. Singh, “Evaluation of bioelectric signals for human recognition,” *Procedia Computer Science*, vol. 48, pp. 746–752, 2015, international Conference on Computer, Communication and Convergence (ICCC 2015).
- [17] J. E. Hall, *Guyton and Hall Textbook of Medical Physiology*, 12th ed. Philadelphia: Saunders, 2010.
- [18] R. Martinek, M. Ladrova, M. Sidikova, R. Jaros, K. Behbehani, R. Kahankova, and A. Kawala-Sterniuk, “Advanced bioelectrical signal processing methods: Past, present and future approach—part i: Cardiac signals,” *Sensors*, vol. 21, no. 15, 2021.
- [19] A. Wood, “Physiology, biophysics, and biomedical engineering.” *Medical Physics*, vol. 41, 2012.
- [20] G. De Polsi, M. Martínez, P. Guiraud, A. López de Lacalle, V. Tejera, E. Catsigeras, M. Barrios, F. Cubría, and P. Lorenzo, *Neurodinámica Determinista*. Montevideo : FACULTAD DE INGENIERIA, 2016.
- [21] H. Ghapanchizadeh, S. A. Ahmad, A. J. Ishak, and M. S. Al-Quraishi, “Review of surface electrode placement for recording electromyography signals,” *Biomedical Research-tokyo*, 2017.
- [22] B. Feinstein, B. Lindegård, E. Nyman, and G. Wohlfart, “Morphologic studies of motor units in normal human muscles.” *Acta anatomica*, vol. 23 2, pp. 127–42, 1955.
- [23] L. F. Bender, “Muscles alive: Their functions revealed by electromyography,” *JAMA*, vol. 201, pp. 277–277, 1967.
- [24] G. L. Soderberg and T. M. Cook, “Electromyography in biomechanics,” *Physical Therapy*, vol. 64, no. 12, pp. 1813–1820, 1984.
- [25] R. E. Burke and P. Tsairis, “Anatomy and innervation ratios in motor units of cat gastrocnemius,” *The Journal of Physiology*, vol. 234, no. 3, pp. 749–765, 1973.

- [26] M. A. Cavalcanti Garcia and T. Vieira, “Surface electromyography: Why, when and how to use it,” *Revista Andaluza de Medicina del Deporte*, vol. 4, pp. 17–28, 2011.
- [27] M. Hakonen, H. Piitulainen, and A. Visala, “Current state of digital signal processing in myoelectric interfaces and related applications,” *Biomedical Signal Processing and Control*, vol. 18, pp. 334–359, 2015.
- [28] M. Verma, “Artificial intelligence and its scope in different areas with special reference to the field of education,” *International Journal of Advanced Educational Research*, vol. 3, pp. 05–10, 2018.
- [29] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, “A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955,” *AI Magazine*, vol. 27, no. 4, p. 12, 2006.
- [30] M. Aceves-Fernandez, *Inteligencia Artificial para Programadores con Prisa*. Amazon self publishing (próximamente ed. Planeta), 10 2021.
- [31] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
- [32] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, “What is machine learning? a primer for the epidemiologist,” *American Journal of Epidemiology*, vol. 188, no. 12, pp. 2222–2239, 2019.
- [33] D. Hand, “Latent variable models and factor analysis: A unified approach, third edition by david j. bartholomew, martin knott, irini moustaki,” *International Statistical Review*, vol. 81, 2013.
- [34] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis : A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, 2018.
- [35] X. Du, Y. Cai, S. Wang, and L. Zhang, “Overview of deep learning,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 159–164.
- [36] S.-C. Wang, *Artificial Neural Network*. Boston, MA: Springer US, 2003, pp. 81–100.
- [37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated feedback recurrent neural networks,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37. PMLR, 2015, pp. 2067–2075.
- [38] C. Arana, “Redes Neuronales Recurrentes: Análisis de los Modelos Especializados en Datos Secuenciales,” Universidad del CEMA, CEMA Working Papers: Serie Documentos de Trabajo. 797, 2021.

- [39] A. Zhang, Z. Lipton, M. Li, and A. Smola, “Dive into deep learning,” 06 2021.
- [40] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, 1994.
- [41] F. Informatik, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” *A Field Guide to Dynamical Recurrent Neural Networks*, 2003.
- [42] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 1997.
- [43] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, vol. 53, 2020.
- [44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.
- [45] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM Comput. Surv.*, vol. 54, 2022.
- [46] H. Arslan, M. Fishel, and G. Anbarjafari, “Doubly attentive transformer machine translation,” 2018.
- [47] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, “Transformers in time series: A survey,” 2022.
- [48] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A time series is worth 64 words: Long-term forecasting with transformers,” 2023.
- [49] E. M. Tamil, N. S. Bashar, M. Y. I. Idris, and A. M. Tamil, “A review on feature extraction & classification techniques for biosignal processing (part iii: Electromyogram),” in *4th Kuala Lumpur International Conference on Biomedical Engineering 2008*. Springer Berlin Heidelberg, 2008, pp. 117–121.
- [50] M. Raez, M. Hussain, and F. Mohd-Yasin, “Techniques of emg signal analysis: Detection, processing, classification and applications,” *Biological Procedures Online*, vol. 8, pp. 11–35, 2006.
- [51] J. Yousefi and A. Hamilton-Wright, “Characterizing emg data using machine-learning tools,” *Computers in Biology and Medicine*, vol. 51, pp. 1–13, 2014.
- [52] A. Ojha, “An introduction to electromyography signal processing and machine learning for pattern recognition: A brief overview,” *Extensive Reviews*, vol. 3, pp. 24–37, 2023.

- [53] 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, 2023.
- [54] C. De Luca, L. Gilmore, M. Kuznetsov, and S. Roy, “Filtering the surface emg signal: Movement artifact and baseline noise contamination,” *Journal of biomechanics*, vol. 43, pp. 1573–9, 2010.
- [55] J. Drake and J. Callaghan, “Elimination of electrocardiogram contamination from electromyogram signals: An evaluation of currently used removal techniques,” *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, vol. 16, pp. 175–87, 2006.
- [56] J. G. Webster, “Reducing motion artifacts and interference in biopotential recording,” *IEEE Transactions on Biomedical Engineering*, vol. BME-31, no. 12, pp. 823–826, 1984.
- [57] G. Naik and R. Marson, “Computational intelligence in electromyography analysis - a perspective on current applications and future challenges,” 2012.
- [58] H. de Talhouet and J. G. Webster, “The origin of skin-stretch-caused motion artifacts under electrodes.” *Physiological measurement*, vol. 17 2, pp. 81–93, 1996.
- [59] E. Clancy, E. Morin, and R. Merletti, “Sampling, noise-reduction and amplitude estimation issues in surface electromyography,” *Journal of Electromyography and Kinesiology*, vol. 12, no. 1, pp. 1–16, 2002.
- [60] A. Vijayvargiya, V. Gupta, R. Kumar, N. Dey, and J. Tavares, “A hybrid wd-eemd semg feature extraction technique for lower limb activity recognition,” *IEEE Sensors Journal*, vol. 21, pp. 20 431–20 439, 2021.
- [61] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.
- [62] H. Obeid, H. Khettab, L. Marais, M. Hallab, S. Laurent, and P. Boutouyrie, “Evaluation of arterial stiffness by finger-toe pulse wave velocity: optimization of signal processing and clinical validation,” *Journal of hypertension*, vol. 35, 2017.
- [63] V. Zschorlich, “Digital filtering of emg-signals,” *Electromyography and clinical neurophysiology*, vol. 29, pp. 81–86, 1989.
- [64] J. Wang, L. Tang, and J. Bronlund, “Surface emg signal amplification and filtering,” *International Journal of Computer Applications*, vol. 82, pp. 15–22, 2013.
- [65] R. Canal, “Comparison of wavelet and short time fourier transform methods in the analysis of emg signals,” *Journal of medical systems*, vol. 34, pp. 91–4, 2010.

- [66] F. A. J. Cortes, and J. Chaves Osorio, “Del análisis de fourier a las wavelets análisis de fourier,” *Scientia Et Technica*, vol. 1, 2007.
- [67] P. Duhamel and M. Vetterli, “Fast fourier transforms: A tutorial review and a state of the art,” *Signal Processing*, vol. 19, pp. 259–299, 1990.
- [68] C. Burrus, *Introduction to Wavelets and Wavelet Transforms: a Primer*. Prentice Hall International, 1998.
- [69] C. E. Heil and D. F. Walnut, “Continuous and discrete wavelet transforms,” *SIAM Review*, vol. 31, pp. 628–666, 1989.
- [70] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005, 1990.
- [71] C. Guarnizo, “Segmentación automática de señales no estacionarias utilizando la transformada wavelet estacionaria,” *Scientia Et Technica*, 2007.
- [72] H. Ashraf, M. Waris, S. Gilani, A. Kashif, M. Jamil, M. Jochumsen, and I. Niazi, “Evaluation of windowing techniques for intramuscular emg-based diagnostic, rehabilitative, and assistive devices,” *Journal of Neural Engineering*, vol. 18, 2020.
- [73] J. Sedlák, D. Špulák, R. Čmejla, R. Bačáková, M. Chrástková, and B. Kračmar, “Segmentation of surface emg signals,” in *2013 International Conference on Applied Electronics*, 2013, pp. 1–4.
- [74] J. D. C. Z. V. Mayorga-Ortiz, P. Valdez-González, “Detección automática y clasificación de eventos en sonidos cardiopulmonares de sujetos saludables,” *Revista Mexicana de Ingeniería Biomédica*, 2018.
- [75] E. Florez, S. Cardona, and L. Jordi, “Selecting the temporal window in the short time fourier transforms used in the vibration signal analysis to determine flats at train’s wheels,” *Revista Facultad de Ingeniería Universidad de Antioquia*, pp. 145–158, 2009.
- [76] A. Ullah, S. Ali, I. U. Khan, M. Khan, and S. Faizullah, “Effect of analysis window and feature selection on classification of hand movements using emg signal,” *Advances in Intelligent Systems and Computing*, pp. 400–415, 2020.
- [77] N. Nazmi, M. A. Abdul Rahman, S.-i. Yamamoto, S. A. Ahmad, M. Malarvili, S. A. Mazlan, and H. Zamzuri, “Assessment on stationarity of emg signals with different windows size during isotonic contractions,” *Applied Sciences*, vol. 7, 2017.
- [78] A. Pozo and R. Trujillo Codorniu, “Ventanas deslizantes por bloques para la implementación online de la transformada wavelet discreta,” *Computación y Sistemas*, vol. 18, 2015.

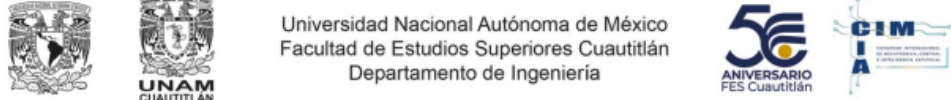
- [79] M. Arif, H. Hassan, D. Nasien, and H. Haron, “A review on feature extraction and feature selection for handwritten character recognition,” *International Journal of Advanced Computer Science and Applications*, vol. 6, 2015.
- [80] A. Turner, D. Shieff, A. Dwivedi, and M. Liarokapis, “Comparing machine learning methods and feature extraction techniques for the emg based decoding of human intention,” vol. 2021, 2021.
- [81] D. Tkach, H. Huang, and T. Kuiken, “Study of stability of time-domain features for electromyographic pattern recognition,” *Journal of neuroengineering and rehabilitation*, vol. 7, p. 21, 2010.
- [82] C. Lin, “Research progress of deep learning based medical image classification techniques,” in *Proceedings of the 2023 4th International Symposium on Artificial Intelligence for Medicine Science*. Association for Computing Machinery, 2024, p. 136–140.
- [83] I. Azuri, I. Goldian, N. Regev-Rudzki, G. Fantner, and S. Cohen, “The role of convolutional neural networks in scanning probe microscopy: a review,” *Beilstein Journal of Nanotechnology*, vol. 12, pp. 878–901, 2021.
- [84] K. Saladin, *Anatomy & Physiology: The Unity of Form and Function*. McGraw-Hill, 2012. [Online]. Available: <https://books.google.com.mx/books?id=Ur6rYgEACAAJ>
- [85] A. F. M. A. G. H. E. Ramírez Ángeles Iván Joel, Razo López Naela Berenice, “Metodología para la adquisición de señales electromiográficas en el brazo utilizando un lector de señales multicanal,” *La Mecánica en México*, vol. 8, pp. 22–36, 2019.
- [86] A. Phinyomark, R. N. Khushaba, and E. Scheme, “Feature extraction and selection for myoelectric control based on wearable emg sensors,” *Sensors*, vol. 18, no. 5, 2018.
- [87] Q. Pan, L. Zhang, G. Dai, and H. Zhang, “Two denoising methods by wavelet transform,” *IEEE Transactions on Signal Processing*, vol. 47, no. 12, pp. 3401–3406, 1999.
- [88] M. Chafii, J. Palicot, and R. Gribonval, “Wavelet modulation: An alternative modulation with low energy consumption,” *Comptes Rendus Physique*, vol. 18, no. 2, pp. 156–167, 2017.
- [89] R. Beniwal, M. Saini, A. Nayyar, B. Qureshi, and A. Aggarwal, “A critical analysis of methodologies for detection and classification of power quality events in smart grid,” *IEEE Access*, vol. 9, pp. 83 507 – 83 534, 2021.

- [90] I. Daubechies, *Ten Lectures on Wavelets*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1992.
- [91] A. A. Orozco, C. Guarnizo, and J. D. Echeverri, “Segmentación automática de señales neurofisiológicas utilizando métodos basados en wavelets,” *Scientia et Technica*, vol. 3, no. 32, 2006.
- [92] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction,” *Journal of Applied Science and Technology Trends*, vol. 1, pp. 56–70, 05 2020.
- [93] S. Dara and P. Tumma, “Feature extraction by using deep learning: A survey,” in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1795–1801.
- [94] P. B. Liashchynskiy and P. Liashchynskiy, “Grid search, random search, genetic algorithm: A big comparison for nas,” *ArXiv*, vol. abs/1912.06059, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:209323928>
- [95] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011.
- [96] S. M. Pizer, J. Hong, J. Vicory, Z. Liu, J. Marron, H. young Choi, J. Damon, S. Jung, B. Paniagua, J. Schulz, A. Sharma, L. Tu, and J. Wang, “6 - object shape representation via skeletal models (s-reps) and statistical analysis,” in *Riemannian Geometric Statistics in Medical Image Analysis*, X. Pennec, S. Sommer, and T. Fletcher, Eds. Academic Press, 2020.
- [97] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, “Multiclass confusion matrix reduction method and its application on net promoter score classification problem,” *Technologies*, vol. 9, 2021.
- [98] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, “Evidence based emergency medicine; part 1: Simple definition and calculation of accuracy, sensitivity and specificity,” *Emergency*, vol. 3, pp. 48–49, 2015.
- [99] J. Cerda and L. Cifuentes, “Uso de curvas roc en investigación clínica: Aspectos teórico-prácticos,” *Revista chilena de infectología*, vol. 29, pp. 138 – 141, 04 2012.
- [100] J. Muschelli, “Roc and auc with a binary predictor: a potentially misleading metric,” *Journal of Classification*, vol. 37, 12 2019.

- [101] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Annals of Data Science*, vol. 9, 04 2022.
- [102] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.
- [103] C. Andrade, “The p value and statistical significance: Misunderstandings, explanations, challenges, and alternatives,” *Indian Journal of Psychological Medicine*, vol. 41, p. 210, 05 2019.
- [104] S. Goodman, “A dirty dozen: Twelve p-value misconceptions,” *Seminars in Hematology*, vol. 45, no. 3, pp. 135–140, 2008, interpretation of Quantitative Research.
- [105] R. Sanchez, “t-student: Usos y abusos.” *Rev. Mex. Cardiol*, vol. 26, pp. 59–61, 2015.

6. Appendices

Publicación de artículo en las Memorias del congreso internacional de mecatrónica, control e inteligencia artificial (CIMCIA) 2024.



Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Cuautitlán
Departamento de Ingeniería

ANIVERSARIO
FES Cuautitlán

CIM
IA
CONGRESO INTERNACIONAL DE
MECATRÓNICA, CONTROL E
INTELIGENCIA ARTIFICIAL


Otorgan la presente

Constancia

**A: Flavio Alfonso Juárez-Castro, Marco Antonio Aceves-Fernández,
Sebastián Salazar-Colores, Jesús Carlos Pedraza-Ortega**


Por su participación como ponente del tema
**Classification of Hand Movements Using EMG
Spectrograms and CNN-Based Architectures**
En el 3º Congreso Internacional de Mecatrónica, Control e Inteligencia
Artificial (CIMCIA) realizado del 6 al 8 de noviembre de 2024.

“POR MI RAZA HABLARÁ EL ESPÍRITU”
Cuautitlán Izcalli, Estado de México, noviembre de 2024.



Folio: 202400561

Dr. David Quintanar Guerrero
Director



Certificado del examen de dominio de la lengua del idioma inglés.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

JUAREZ CASTRO FLAVIO ALFONSO

Presentó el **Examen de Manejo de la Lengua** efectuado el día trece de noviembre de dos mil veinticuatro, en el cual obtuvo la siguiente calificación:

9-

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día veintidós de noviembre de dos mil veinticuatro.



Atentamente,
"Enlazar Culturas por la Palabra"


DRA. MA. DE LOURDES RICO CRUZ

MLRC/mgoa*CL*FLL-C.-382

UAQ ORGANISMO DE
SEGURIDAD

01 442 211
01 442 192 12 00 EXT. 41018

Campus Aeropuerto, Avda. Vial Frey Joaquín Soria S/N,
Santiago de Querétaro, Qro. México. C.P. 76100

FOLIO: 65206

Certificado del examen de comprensión de textos en inglés.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

JUAREZ CASTRO FLAVIO ALFONSO

Presentó y acreditó el **Examen de Comprensión de Textos en Inglés** efectuado el día nueve de octubre de dos mil veinticuatro.

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día diecisiete de octubre de dos mil veinticuatro.



Atentamente,
"Enlazar Culturas por la Palabra"


DRA. MA. DE LOURDES RICO CRUZ

MLRC/mgoa*CL*FLL-C.-155

UAQ

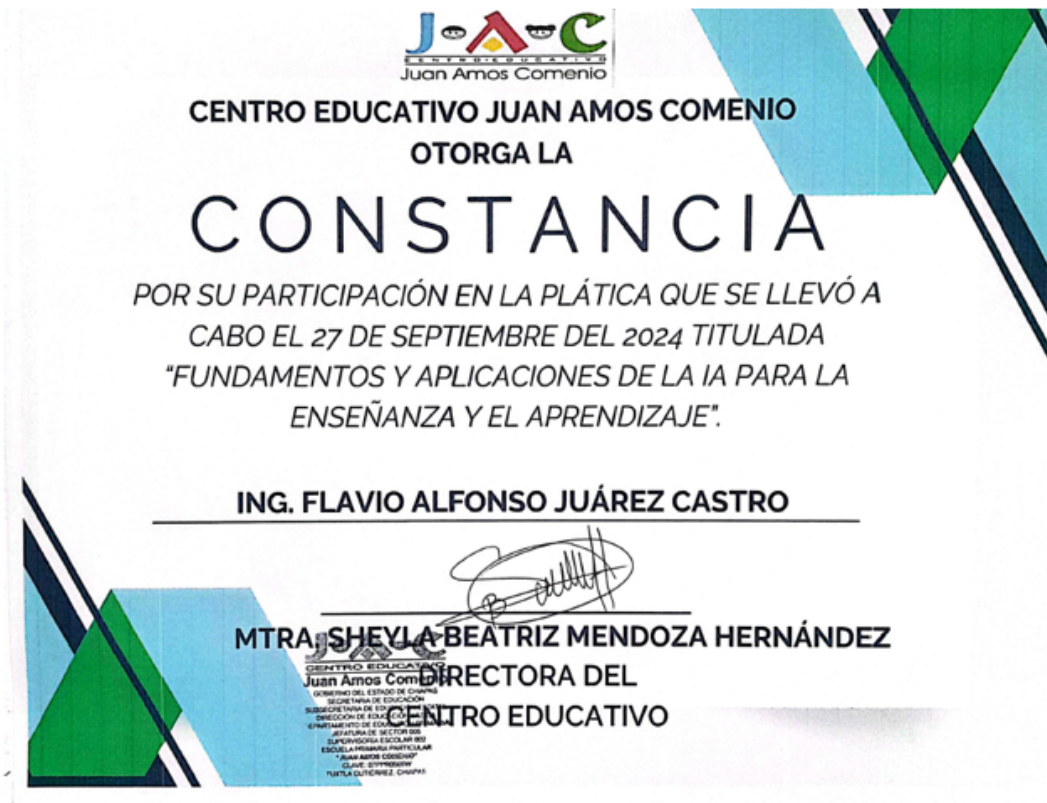
UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

☎ 01 442 192 12 00 EXT. 41010


📍 Campus Aeropuerto, Avda. Valero Jiménez Suro 150,
Santiago de Querétaro, Qro. México, C.P. 76142

FOLIO: 65433

Constancia de participación en la presentación 'Fundamentos y aplicaciones de la IA para la enseñanza y el aprendizaje' al cuerpo docente del centro educativo Juan Amos Comenio.



Constancia de participación como parte del equipo organizador del XVIII Coloquio de posgrado de la facultad de ingeniería en la Universidad Autónoma de Querétaro.



UNIVERSIDAD
AUTÓNOMA
DE QUERÉTARO



FACULTAD
DE INGENIERÍA



DIFI
POSGRADO



COLOQUIO
18^º
POSGRADO FI


Se otorga la presente
CONSTANCIA a:

Flavio Alfonso Juárez Castro

Por su participación en el
XVIII Coloquio de Posgrado de la Facultad de Ingeniería
de la Universidad Autónoma de Querétaro como parte del equipo organizador

*"La tecnología solo avanza cuando
se nutre del conocimiento científico"*

Noviembre de 2024
Facultad de Ingeniería



Dra. María de la Luz Pérez Rea
Directora
Facultad de Ingeniería



Dr. Juan Carlos Jáuregui Correa
Jefe de la División de Investigación y Posgrado
Facultad de Ingeniería