



Evaluación automática de la enfermedad de Parkinson a partir de señales de escritura online considerando arquitecturas de aprendizaje profundo como las CNNs

Julio César Zapata Ortiz

Trabajo de grado presentado para optar al título de Ingeniero Electrónico

Asesor

Juan Rafael Orozco Arroyave, Prof. Dr.-Ing.

Coasesor

Daniel Escobar Grisales, Magíster (MSc) en Telecomunicaciones

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2023

Cita	Zapata Ortiz [1]
Referencia Estilo IEEE (2020)	[1] J.C Zapata Ortiz, “Evaluación automática de la enfermedad de Parkinson a partir de señales de escritura online considerando arquitecturas de aprendizaje profundo como las CNNs”, Trabajo de grado profesional, Ingeniería Electrónica, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2023.



Grupo de Investigación Telecomunicaciones Aplicadas (GITA).



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Augusto Enrique Salazar Jiménez.

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

Agradecimientos

En primer lugar, quiero agradecer a mis padres Julio Zapata T. y Rosa Ortiz, quienes me han brindado apoyo incondicional para cumplir con mis objetivos académicos a pesar de las adversidades que presenta la vida, además, agradecer a mi hermana Yenifer Zapata por sus consejos y confianza depositada en mí al ayudarme a conseguir el soporte económico a lo largo de mi carrera para poder concentrarme en el estudio. También, agradecer a mi pareja Natalia Machado que ha estado conmigo en todo el pregrado, especialmente en los últimos semestres su apoyo incondicional fue muy importante para mí. Ser foráneo es un estado algo complicado de sobrellevar, pero ellos siempre han estado brindándome compañía sin importar la distancia.

Agradezco a la Universidad de Antioquia por todos los apoyos que me ha brindado desde que ingresé, por todos los conocimientos y experiencias adquiridas en todos sus espacios tanto académicos como sociales y culturales. Además, por permitirme conocer a mi grupo de amigos que a pesar de no estar tanto tiempo juntos los aprecio, en especial a mis amigos Jeferson Gallo, Camilo Anacona y Felipe Carmona, que lo han sido desde el primer semestre. Tardé en empezar un proceso formativo con GITA como lo hablamos en algunas conversaciones antes y durante la Pandemia con Jeferson, pero más vale tarde que nunca.

Finalmente, quiero agradecer a mi asesor Rafael Orozco y coasesor Daniel Escobar por la confianza, paciencia y enseñanzas en todo el proceso para la realización del trabajo de grado que para mí significa el primer escalón alcanzado de un largo camino por recorrer en la investigación en el que tengo emoción de enfrentarme a nuevos retos. Con este trabajo se materializa uno de mis sueños que tanto anhele y al observar en retrospectiva por todo lo afrontado puedo decir orgullosamente que logré culminar con este hermoso proceso, por ello agradezco a todos y cada uno de ustedes por lo que me han brindado, gracias.

TABLA DE CONTENIDO

RESUMEN.....	8
ABSTRACT.....	9
1. INTRODUCCIÓN.....	10
2. OBJETIVOS.....	14
2.1. Objetivo general.....	14
2.2. Objetivos específicos.....	14
3. MARCO TEÓRICO.....	15
3.1. Algoritmos de aprendizaje automático.....	15
3.2. Deep Learning.....	15
3.2.1. Redes neuronales artificiales.....	16
3.2.2. Perceptrón simple.....	16
3.2.3. Perceptrón multicapa.....	17
3.2.4. Backpropagation.....	18
3.3. Algoritmos de optimización.....	20
3.3.1. Optimizador descenso del gradiente estocástico (SGD por sus siglas en inglés).....	21
3.3.2. Optimizador Adam (Adaptative Moment Estimation).....	21
3.4. Estrategias de regularización.....	22
3.4.1. Parámetro de regularización $L2$	22
3.4.2. Early stopping.....	22
3.4.3. Dropout.....	23
3.5. Redes neuronales convolucionales.....	23
3.5.1. Capas convolucionales.....	23
3.5.2. Capas de agrupación.....	25
3.5.3. Capa totalmente conectada.....	25

3.6.	CNNs Unidimensionales	26
3.7.	Estrategias de validación	30
3.7.1.	Validación cruzada K-fold.....	30
3.8.	Métricas de desempeño	31
3.8.1.	Matriz de confusión	31
3.8.2.	Curva ROC.....	33
3.8.3.	Área bajo la curva ROC (AUC).....	33
4.	BASE DE DATOS.....	34
5.	METODOLOGÍA	34
5.1.	Experimentos	37
5.1.1.	Optimización de arquitectura monocanal.	37
5.1.2.	Validación cruzada del monocanal	39
5.1.3.	Optimización de arquitectura multicanal	39
5.1.4.	Validación cruzada del multicanal	39
6.	RESULTADOS.....	40
6.1.	Optimización de la arquitectura monocanal	40
6.1.1.	Análisis del desempeño en las distintas resoluciones temporales	43
6.2.	Validación cruzada del modelo monocanal.....	43
6.3.	Optimización de la arquitectura multicanal.....	44
6.4.	Validación cruzada del multicanal	45
6.5.	Comparación de las métricas de desempeño de la validación cruzada en los modelos monocanal y multicanal	46
7.	CONCLUSIONES	47
8.	REFERENCIAS.....	49

LISTA DE TABLAS

TABLA I. Información demográfica y clínica de los sujetos. μ : media, σ : desviación estándar..	34
TABLA II. Resultados de los mejores parámetros para la arquitectura 1D-CNN con el modelo monocanal. Res. Temp : Resolución temporal, KS : Kernel y Stride, Solp : Solapamiento, LR : Tasa de aprendizaje, Filt : número de filtros iniciales, FC : capas completamente conectadas y Dpo : Dropout.....	41
TABLA III. Métricas de desempeño por segmentos para los mejores parámetros del modelo monocanal	41
TABLA IV. Métricas de desempeño por sujeto para los mejores parámetros del modelo monocanal	41
TABLA V. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo monocanal con los segmentos. μ : media, σ : desviación estándar	44
TABLA VI. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo monocanal por sujeto. μ : media, σ : desviación estándar	44
TABLA VII. Métricas de desempeño en la clasificación por segmentos y por sujeto en el modelo multicanal tomando todas las señales dinámicas	45
TABLA VIII. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo multicanal tomando todas las señales. μ : media, σ : desviación estándar.....	46
TABLA IX. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo multicanal tomando las señales X, Z y Altitud. μ : media, σ : desviación estándar	46

LISTA DE FIGURAS

Fig. 1. Perceptrón multicapa	17
Fig. 2. Capas de agrupación comunes.....	25
Fig. 3. Arquitectura de una 1D-CNN.....	26
Fig. 4. Reducción de dimensionalidad en una señal dinámica en una 1D-CNN.....	28
Fig. 5. validación cruzada k-fold con $k=5$	30
Fig. 6. Ejemplo de matriz de confusión.	31
Fig. 7. Curva ROC y AUC.....	33
Fig. 8. Metodología general.	35
Fig. 9. Distribución de la duración en la prueba para todos los sujetos.	36
Fig. 10. Metodología para el modelo monocanal.....	38
Fig. 11. Metodología para el modelo multicanal.	40
Fig. 12. Curva ROC y distribución de scores para el mejor resultado del modelo monocanal	42
Fig. 13. Curvas ROC para los modelos monocanal.	42
Fig. 14. Exactitudes obtenidas al varias la resolución temporal en los modelos monocanal.....	43
Fig. 15. Curva ROC y distribución de scores para el mejor modelo multicanal.....	45

RESUMEN

La enfermedad de Parkinson es una enfermedad neurodegenerativa caracterizada por la aparición de diversos trastornos cognitivos y motores, provocados por la degeneración de las neuronas dopaminérgicas. Es una de las enfermedades neurodegenerativas más comunes y su diagnóstico es generalmente costoso en tiempo y dinero. Entre los síntomas motores más destacados está el deterioro en la calidad de la escritura donde síntomas como la micrografía y la disgrafía empiezan a manifestarse en etapas tempranas de la enfermedad, debido a esto es posible evaluar automáticamente la enfermedad de Parkinson analizando la pérdida de habilidades grafomotoras cuando el paciente escribe en una tableta digitalizadora que captura seis señales diferentes en forma de series temporales que son las coordenadas cartesianas (x, y, z), ángulo azimut, ángulo de altitud y presión.

En este trabajo se consideran las tareas de escritura online de 55 pacientes con la enfermedad de Parkinson y 55 personas sanas. Particularmente se trabajó con dibujos de la espiral de Arquímedes con el fin de analizar las diferentes señales extraídas cuando un sujeto escribe, dichas señales fueron utilizadas para ajustar arquitecturas de aprendizaje profundo basadas en redes neuronales convolucionales unidimensionales, donde cada señal es analizada de forma independiente usando un enfoque monocanal con el fin de identificar las señales que mejor discriminan la escritura de los pacientes con enfermedad de Parkinson. Además, se evalúa la combinación de las señales usando un enfoque multicanal, donde se busca determinar si se obtiene un modelo más robusto al analizar todas o algunas de las señales de manera conjunta. La generalización de los modelos desarrollados fue verificada mediante una estrategia de validación cruzada donde los mejores resultados se obtienen usando la señal de altitud para el enfoque monocanal y combinando las señales x, z y altitud en el enfoque multicanal. El modelo monocanal presenta un mejor balance en la clasificación de ambas clases. Finalmente, el multicanal presenta un desempeño inferior esto puede deberse a que algunas señales no aportan más información para detectar la enfermedad de Parkinson por medio del deterioro de las habilidades grafomotoras.

***Palabras clave* —Enfermedad de Parkinson, Redes Neuronales Convolucionales Unidimensionales, Escritura online, Señales Dinámicas, Series Temporales.**

ABSTRACT

Parkinson's disease is a neurodegenerative disease characterized by the appearance of various cognitive and motor disorders caused by the degeneration of dopaminergic neurons. It is one of the most common neurodegenerative diseases and its diagnosis is often expensive in time and money. Among the most prominent motor symptoms is the decline in the quality of handwriting where symptoms such as micrographia and dysgraphia appear in early stages of the disease, due to this it is possible to automatically evaluate Parkinson's disease by analyzing the loss of skills graphomotor when the patient writes on a digital tablet that captures six different signals in the form of time series which are Cartesian coordinates (x, y, z), azimuth angle, altitude angle and pressure.

In this work, the online handwriting tasks of 55 patients with Parkinson's disease and 55 healthy people are considered. Archimedean spiral drawings were used to analyze the different signals extracted when a subject draws, these are used to adjust deep learning architectures based on one-dimensional convolutional neural networks, where each signal is analyzed independently using a single-channel approach to identify the signals that best discriminate the handwriting of Parkinson's disease patients. In addition, the combination of the signals is evaluated using a multichannel approach, the objective is to determine whether a more robust model is obtained by analyzing all or some of the signals together. The generalization of the developed models was verified by a cross-validation strategy where the best results are obtained by using the altitude signal for the single-channel approach and combining the x, z and altitude signals in the multichannel approach. The single-channel model yields a better balance in the classification of both classes. Finally, the multichannel presents an inferior performance this may be because some signals do not provide more information to detect Parkinson's disease through the impairment of graphomotor skills.

Keywords — Parkinson's Disease, One-Dimensional Convolutional Neural Networks, Online Handwriting, Dynamic Signals, Time Series.

1. INTRODUCCIÓN

La Enfermedad de Parkinson (EP) es el segundo trastorno neurodegenerativo más común en la población mundial, en donde 6 millones de personas padecen esta enfermedad [1] y se espera que 14.2 millones de personas se vean afectadas por la EP para el año 2040 [2]. Algunos síntomas y signos motores en las personas que padecen la EP son bradicinesia, rigidez, temblor en reposo e inestabilidad postural [1]. Los síntomas motores provienen de la pérdida progresiva de neuronas dopaminérgicas, aunque se reconoce que la patología en la EP no se limita a este tipo de neuronas [3]. El deterioro en la escritura es uno de los signos iniciales que motiva a los pacientes a una primera consulta con un médico [4], [5]; recientemente diversos biomarcadores han sido objeto de investigación, debido a que, sería posible diagnosticar la EP en una etapa más temprana y por consiguiente realizar un tratamiento adecuado para controlar la progresión de la enfermedad [4].

En etapas tempranas de la EP comienzan a manifestarse deterioros en la calidad de la escritura, entre los más comunes esta la micrografía, velocidad, fluidez, aceleración y decadencia en la precisión [2], [5]. Por lo tanto, el deterioro en las habilidades grafomotoras de los pacientes con EP puede ser usado como biomarcador para apoyar el diagnóstico de la enfermedad de una manera precisa, automática y poco intrusiva [6].

Diversos estudios han considerado y trabajado con escritura online, donde las tareas de escritura se realizan mediante tabletas digitalizadoras o bolígrafos inteligentes biométricos que logran capturar la información dinámica que se desarrolla en el proceso de escritura [2]. En Ríos-Urrego et al. [6] evaluaron el rendimiento de características cinemáticas, geométricas y de dinámica no lineal para clasificar pacientes con EP y sujetos de control sanos (CS). Los autores consideraron 39 pacientes con EP con una edad promedio de 60 años, un conjunto de CS compuesto por 39 sujetos con una edad promedio de 63 años y un segundo conjunto de CS compuesto de 40 sujetos con una edad promedio de 24 años. Adicionalmente se incluye un conjunto de 12 participantes (6 pacientes con EP y 6 CS) para validar la capacidad de generalización del modelo. Las tareas usadas fueron espirales de Arquímedes y la oración “El abecedario es a, b, c, ..., z”, donde usaron enfoques de aprendizaje clásico como máquinas de soporte vectorial (SVM, por sus siglas en inglés), K vecinos más cercanos (KNN, por sus siglas en inglés) y bosques aleatorios. Usando las tareas de la

espiral y el conjunto de características cinemáticas logran un 94% de precisión en la clasificación del grupo de pacientes con EP y los CS con edad promedio de 60 años y de 87% entre el grupo de pacientes con EP y los CS con edad promedio de 23 años, en el grupo usado para validar la capacidad de generalización se alcanzó una precisión de hasta un 83.3%.

Por otra parte, en Gupta et al. [7] desarrollaron métodos de clasificación de pacientes con EP específico por sexo y dependiente de la edad mediante escritura online. Usaron el conjunto de datos "Parkinson's Disease Handwriting Database" (PaHaw) que consta de 37 pacientes con EP (19 hombres y 18 mujeres; $69,3 \pm 10,9$ años) y 38 controles (20 hombres y 18 mujeres; $62,4 \pm 11,3$ años). Todos los sujetos realizaron siete tareas de escritura que consistieron en letras cursivas o bi/trigramas de letras (Tarea 1-3), escritura de un trazo largo (Tarea 4-6) y escritura de una oración más larga para capturar fatiga (Tarea 7). Entre las señales capturadas por la tableta están: coordenadas cartesianas (x, y, z), marca de tiempo, estado del botón, presión e inclinación. A partir de las diferentes señales capturadas, se extraen características cinemáticas, entrópicas y energéticas. Una SVM es usada para generar cuatro modelos diferentes: generalizado (sin información previa de sexo o edad), específico por sexo, dependiente de la edad y dependiente de sexo-edad. Los mejores resultados son obtenidos con la clasificación específica para mujeres, donde se logra una precisión de hasta 83.75%, superando al clasificador dependiente de edad en 4% y al clasificador generalizado en hasta 8%. Los autores concluyen que usar la información de la edad y el género ayuda a mejorar la precisión del modelo en comparación con el clasificador generalizado.

Recientemente algunos trabajos han abordado enfoques de aprendizaje profundo. En Khatamino et al. [8] usan una red neuronal convolucional (CNN) para el diagnóstico de EP con pruebas manuscritas de espirales estáticas y dinámicas, donde la primera se basa en dibujar una espiral de Arquímedes y la segunda es dibujar la misma espiral, pero esta no tiene una posición fija en la pantalla y desaparece en intervalos predeterminados. La base de datos se denomina "Parkinson HW" y cuenta con 57 pacientes con EP y 15 sujetos de CS. En los dibujos de espirales realizados extraen señales dinámicas como las coordenadas cartesianas (x, y, z), presión, ángulo de agarre y marca de tiempo; estas señales son usadas para generar imágenes con la finalidad de usarlas como entrada a otra CNN, además de la CNN cuya entrada son los dibujos de las espirales.

Obtienen una exactitud de 88% en ambos enfoques, usando los dibujos de espirales dinámicas y las imágenes generadas a partir de las señales dinámicas. En Chakraborty et al. [9] buscan clasificar pacientes con EP y sujetos CS a partir de dos tareas de escritura, espirales y ondas. Los datos utilizados fueron tomados de una base de datos de Kaggle, que consta de 27 pacientes con EP y 27 sujetos de CS. Dos arquitecturas basadas en CNNs son usadas, una para cada tarea y debido a que el conjunto de imágenes es reducido, los autores consideraron diferentes estrategias de aumento de datos como rotaciones, volteos, cambios de brillo, entre otros. El modelo completo se entrenó con los datos de 55 sujetos y alcanzó una exactitud general de 93,3%. En Pereira et al. [10] con la ayuda de un bolígrafo inteligente biométrico capturan la información de escritura para crear un conjunto de datos, donde las tareas eran dibujar círculos en el papel, en el aire, dibujar espirales, meandros y movimientos de ambas muñecas. El conjunto de datos se compone de 74 pacientes con EP y 18 sujetos de CS. El bolígrafo inteligente biométrico tiene seis sensores que capturan señales de micrófono, agarre de los dedos, presión axial de recarga de tinta e inclinación-aceleración en las coordenadas cartesianas (x, y, z). Las señales obtenidas son mapeadas a una imagen donde cada tarea se compone de n filas (tiempo de la tarea en milisegundos) y 6 columnas (representan las señales de los sensores), posteriormente, transforman esta información en una matriz cuadrada y la normalizan para manejar las imágenes en escala de grises. Del proceso de mapeado consideran dos resoluciones distintas, las cuales son usadas como entrada a dos arquitecturas de CNNs diferentes, ImageNet y CiFar-10. Los autores evalúan ambos modelos utilizando dos enfoques: un enfoque de evaluación única, donde las tareas no son combinadas y otro enfoque donde todas las tareas son combinadas. Usando la arquitectura ImageNet y el enfoque de tareas combinadas logran una exactitud de hasta 93.5% para discriminar entre pacientes con EP y controles. En Kamran et al. [11] los autores usan 4 conjuntos de datos de escritura offline: PaHaw, HandPD, NewHandPD y Parkinson's Drawings; esto con el fin de clasificar pacientes con EP y sujetos de CS usando una arquitectura basada en CNNs. En este trabajo fueron consideradas diferentes técnicas de aprendizaje por transferencia y estrategias de aumento de datos. Los mejores resultados fueron obtenidos usando la arquitectura AlexNet ajustada y la estrategia de aumento de datos basada en los cambios de iluminación, donde se obtuvieron resultados de hasta 99.2% superando en hasta un 8% los enfoques basados en aprendizaje por transferencia. Aunque en este trabajo se usa escritura offline la estrategia de aumento de datos puede ser aplicable en un conjunto de datos de escritura

online, esto es con el fin de brindar más información al modelo de tal forma que este pueda ajustarse mejor y clasificar correctamente nuevas entradas como se logró en este trabajo.

Con el fin de explotar la información temporal que brinda la tableta en la escritura online, algunos enfoques consideran arquitecturas recurrentes como las Bidirectional Long-Short Term Memory (Bi-LSTM). En Taleb et al. [12] los autores usan series temporales como las coordenadas cartesianas, presión del lápiz, marca de tiempo, ángulo de altitud y azimut. Estas señales son generadas realizando repetidamente tareas de escritura como letra “l” cursiva, onda triangular y rectangular, “Monday”, “Tuesday”, nombres y apellidos. La información la obtienen de 21 pacientes con EP y 21 sujetos de CS del conjunto de datos HandPDMultiMC. Además, mapean cada serie temporal individualmente como una imagen de espectrograma usando la Transformada de Fourier de Tiempo Reducido. En este trabajo diseñaron una arquitectura basada en CNNs que recibe como entrada las imágenes mapeadas de las series temporales de forma independiente y una CNN Bi-LSTM cuya entrada son las series temporales crudas sin generar subsecuencias más cortas, su arquitectura consta de capas de CNNs unidimensionales para la extracción de características y Bi-LSTM al final para analizar las series temporales. Los experimentos en ambos enfoques son: evaluación única y una evaluación conjunta, donde se toma el número de series temporales como hiperparámetro de manera que devuelva la mayor exactitud general. Para CNNs con imágenes de espectrogramas la mejor combinación de series temporales fue $x+y+z+$ presión+altitud. En cuanto a CNN Bi-LSTM el mejor desempeño fue con $x+y+z+$ presión+altitud+azimut, donde ambos modelos logran una precisión de hasta 83%. Los autores destacan la clasificación de EP vs controles mediante aprendizaje profundo como una tarea desafiante, debido a disponibilidad de datos limitada. En un trabajo más reciente de Taleb et al. [13], los autores buscan reducir el sobreajuste de los modelos usando estrategias de aumento de datos y de transferencia de aprendizaje. Los mejores modelos presentados en [12], CNNs y CNN-Bi-LSTM, son usados para clasificar pacientes con EP de sujetos de CS usando la base de datos HandPDMultiMC. Las técnicas de aumento de datos sobre las series temporales fueron jittering, escalado, distorsión temporal y generación de datos sintéticos. Entre los enfoques de aprendizaje por transferencia, los modelos base son ajustados con la base de datos PaHaw, lo cual provoca la eliminación de la coordenada z porque PaHaw no la incluye. La exactitud más alta fue de 97.62% al combinar todas las tareas considerando la estrategia de aumento de datos basada en el jittering para ajustar la CNN-

BiLSTM. Los autores mencionan que los enfoques de aprendizaje por transferencia no logran buenos resultados debido a la ausencia de la coordenada z.

De acuerdo con la revisión de la literatura, la mayoría de los trabajos han abordado el problema realizando un análisis a la imagen que puede ser generada a partir de las señales que captura el bolígrafo inteligente o la tableta digitalizadora. Otros enfoques en donde se explote la información temporal de las señales han sido poco explorados. En este trabajo se propone realizar la clasificación automática de pacientes con EP vs CS usando arquitecturas de CNNs unidimensionales, de tal manera que se pueda evaluar si el modelo logra obtener la capacidad de discriminación entre clases, usando las señales crudas extraídas en el proceso de escritura online. Las señales extraídas de la tableta serán evaluadas de forma independiente, modificando la resolución temporal de la ventana de análisis, con el fin de determinar la ventana óptima para el análisis de las señales. Luego una metodología basada en CNNs unidimensionales multicanal será usada para combinar la información de las diferentes señales extraídas por la tableta. Finalmente, ambos enfoques serán comparados con el fin de determinar las señales que mejor discriminan la EP y si estas señales son complementarias y permiten obtener un modelo más robusto.

2. OBJETIVOS

2.1. *Objetivo general*

Diseñar e implementar una arquitectura basada en redes neuronales convolucionales unidimensionales con enfoque monocanal y multicanal para clasificar pacientes con enfermedad de Parkinson a partir de señales de escritura online.

2.2. *Objetivos específicos*

- Evaluar el desempeño de arquitecturas basadas en CNNs unidimensionales para la clasificación de pacientes con enfermedad de Parkinson analizando de manera independiente las señales generadas en el proceso de escritura online.

- Evaluar y comparar el desempeño de los modelos basados en CNNs unidimensionales para la clasificación de pacientes con enfermedad de Parkinson considerando ventanas de análisis con diferentes resoluciones temporales.
- Implementar y evaluar modelos de aprendizaje multicanal basados en CNNs unidimensionales con el fin de combinar las diferentes señales extraídas en el proceso de escritura online.
- Aplicando algoritmos en Python y considerando los niveles de acierto correspondientes, se compararán los modelos multicanal con los modelos monocanal considerando diferentes métricas de desempeño; con el fin de determinar si la información de las diferentes señales capturadas es complementaria o si por el contrario son redundantes.

3. MARCO TEÓRICO

3.1. Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático pueden ser supervisados y no supervisados; la principal diferencia entre estos es la existencia de etiquetas en el subconjunto de entrenamiento [14]. En el aprendizaje supervisado el algoritmo observa algunos pares de entrada-salida a manera de ejemplo, con el fin de aprender alguna función que asigna una salida según la entrada [15]. Por otra parte, el aprendizaje no supervisado busca reconocer patrones en los datos sin información previa acerca de las categorías a la que pertenece cada muestra [14]. Aunque existe una gran variedad de algoritmos tradicionales de aprendizaje supervisado, las máquinas de soporte vectorial (SVMs por sus siglas en inglés) han sido ampliamente usadas, principalmente por su alta precisión en espacios con alta dimensionalidad [14], [15].

3.2. Deep Learning

Deep Learning o aprendizaje profundo, es una técnica de aprendizaje automático en donde el modelo implementado aprende a realizar tareas de clasificación directamente de las imágenes, texto o sonido; es decir de la información cruda. Los modelos normalmente se entrenan con un

gran conjunto de datos etiquetados y usando arquitecturas de redes neuronales. La mayoría de los métodos de aprendizaje profundo utilizan arquitectura de redes neuronales profundas, este último término se refiere a la cantidad de capas ocultas en la red neuronal. Uno de los tipos más usados de redes neuronales profundas se conoce como redes neuronales convolucionales (CNNs) [16].

Los tres factores relevantes de las CNNs que las hacen populares para usar en diversos trabajos son: primer factor es eliminar la extracción manual de características, puesto que estas son aprendidas directamente por las CNNs. El segundo factor es producir resultados de clasificación precisos y el último factor es que las CNNs se pueden volver a entrenar para nuevas tareas de reconocimiento, esta estrategia es conocida como aprendizaje por transferencia [17]. La red neuronal usa la estrategia de backpropagation y descenso del gradiente para actualizar los pesos, este proceso se realiza de forma iterativa [18].

3.2.1. Redes neuronales artificiales

Es un sistema computacional que usa una red de diversas funciones para comprender un conjunto de datos entrantes para generar una salida deseada y hace parte de los algoritmos de aprendizaje automático y puede usarse como un elemento en otros algoritmos como las redes neuronales convolucionales. Las redes neuronales están inspiradas en la forma en que las neuronas del cerebro humano funcionan en conjunto para comprender el entorno [19]. En estas redes los nodos que la componen tienen conexiones sinápticas que son direccionales, es decir la información se propaga en un único sentido, en general las neuronas que componen estas redes se agrupan en unidades estructurales denominadas capas, donde el conjunto de una o más capas constituye una red neuronal, redes monocapa o redes multicapa respectivamente. Estas capas son capa de entrada, capas ocultas y capa de salida [21].

3.2.2. Perceptrón simple

Un perceptrón es la unidad básica dentro de una red neuronal y es usado en clasificaciones binarias donde define una recta para determinar la región de decisión para separar dos regiones en el espacio. Por tanto, el perceptrón tiene limitaciones ya que solo puede representar funciones

linealmente separables. La ecuación (1) resume el funcionamiento del perceptrón, donde f es una función de activación, x el conjunto de entradas (características), w los pesos sinápticos, θ el umbral de decisión, y la salida y n el número de entradas.

$$y = f \left(\sum_{j=1}^n w_j x_j - \theta_j \right) \quad (1)$$

3.2.3. Perceptrón multicapa

Debido a las limitaciones presentes en el perceptrón en la que solo puede discriminar patrones mediante un hiperplano lineal (recta), una manera de solucionar y superar esta limitación es añadiendo capas ocultas completamente conectadas obteniendo una red neuronal llamada perceptrón multicapa (MLP por sus siglas en inglés). Esta red se suele entrenar con el algoritmo de backpropagation [21]. En la **Fig. 1** se encuentra un ejemplo de un MLP, donde el número de capas es de 2 puesto que la capa de entrada no implica cálculo alguno.

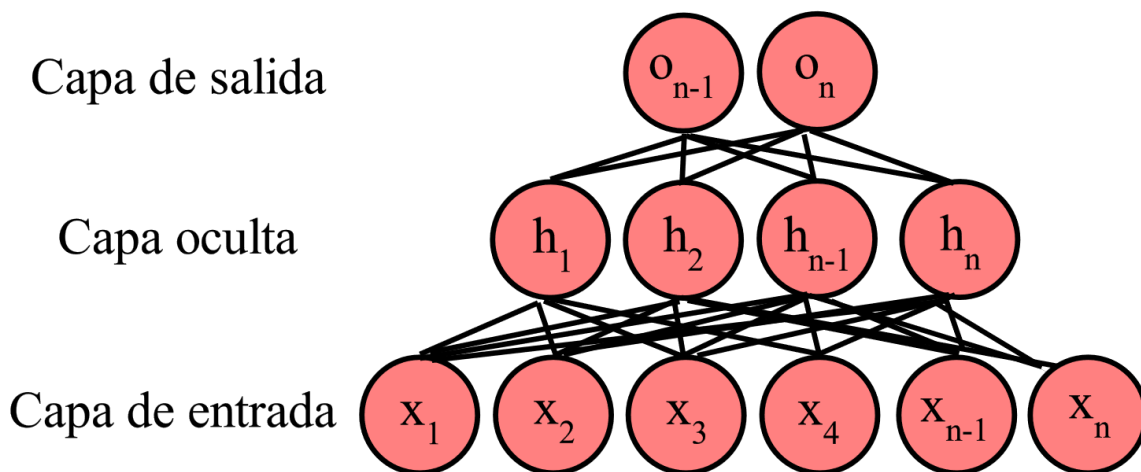


Fig. 1. Perceptrón multicapa

3.2.4. Backpropagation

Es un algoritmo que se usa para entrenar de forma efectiva una red neuronal a través del método de regla de la cadena. Es decir, primero la red va hacia adelante realizando el proceso de clasificación y al terminar todo el recorrido por la arquitectura se devuelve ajustando los pesos de acuerdo con la función de costo. El nivel de ajuste está determinado por los gradientes de la función de costo con respecto a los parámetros [21].

Para explicar este algoritmo se usa el MLP de la **Fig. 1** como guía, además de asumir que la capa oculta no incluye el término de sesgo para simplificar los cálculos. Entonces, primero se realiza la propagación hacia delante para la red neuronal desde la capa de entrada hasta la capa de salida, primero se define la variable intermedia \mathbf{z} en la ecuación (2) siendo esta la salida que genera la capa oculta, donde $\mathbf{W}^{(1)}$ son los pesos de la capa oculta y \mathbf{x} el conjunto de entradas.

$$\mathbf{z} = \mathbf{W}^{(1)}\mathbf{x} \quad (2)$$

Después de ejecutar la variable intermedia \mathbf{z} a través de la función de activación ϕ se obtiene el vector de activación oculto \mathbf{h} , que también es una variable intermedia.

$$\mathbf{h} = \phi(\mathbf{z}) \quad (3)$$

Asumiendo que $\mathbf{W}^{(2)}$ son los pesos correspondientes a la capa de salida, se obtiene la salida resultante de esta capa como:

$$\mathbf{o} = \mathbf{W}^{(2)}\mathbf{h} \quad (4)$$

Ahora, suponiendo que función de pérdida es l y la clase a predecir es y , se calcula el término de pérdida L .

$$L = l(\mathbf{o}, y) \quad (5)$$

Finalmente, se obtiene la función de costo J al añadir un término de regularización de regularización s a (5), que este a su vez internamente contiene un hiperparámetro λ . De esta forma se obtiene:

$$J = L + s \quad (6)$$

Ahora es posible aplicar la propagación hacia atrás con el cálculo de la regla de la cadena donde el algoritmo almacena las variables intermedias (derivadas parciales) requeridas mientras calcula el gradiente con respecto a algunos parámetros. Su funcionamiento es el siguiente, suponiendo que se tiene $\mathbf{Y} = f(\mathbf{X})$ y $\mathbf{Z} = g(\mathbf{Y})$ usando esta regla se puede calcular la derivada parcial de \mathbf{Z} con respecto a \mathbf{X} así:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{X}} = \text{prod} \left(\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}}, \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \right) \quad (7)$$

Donde se usa el operador prod para multiplicar los argumentos después de otras operaciones necesarias como transpuesta, intercambio de posiciones de entrada, esto con la finalidad de simplificar el cálculo. Entonces, el objetivo de backpropagation es calcular los gradientes $\partial J / \partial \mathbf{W}^{(1)}$ y $\partial J / \partial \mathbf{W}^{(2)}$, por lo que se aplica la regla de la cadena y calculamos con (7). El primer paso es calcular los gradientes de la función de costo respecto a L y s .

$$\frac{\partial J}{\partial L} = 1 \text{ y } \frac{\partial J}{\partial s} = 1 \quad (8)$$

Ahora se calcula el gradiente de la función objetivo con respecto a la variable de la capa de salida \mathbf{o} con la regla de la cadena:

$$\frac{\partial J}{\partial \mathbf{o}} = \text{prod} \left(\frac{\partial J}{\partial L}, \frac{\partial L}{\partial \mathbf{o}} \right) \quad (9)$$

A continuación, se calcula los gradientes del término de regularización con respecto a ambos parámetros de pesos de la capa oculta y de salida.

$$\frac{\partial s}{\partial \mathbf{W}^{(1)}} = \lambda \mathbf{W}^{(1)} \text{ y } \frac{\partial s}{\partial \mathbf{W}^{(2)}} = \lambda \mathbf{W}^{(2)} \quad (10)$$

Ahora se procede a calcular $\partial J/\partial \mathbf{W}^{(2)}$ de los parámetros del modelo más cercanos a la capa de salida. Al usar (7) se obtiene:

$$\frac{\partial J}{\partial \mathbf{W}^{(2)}} = \text{prod} \left(\frac{\partial J}{\partial \mathbf{o}}, \frac{\partial \mathbf{o}}{\partial \mathbf{W}^{(2)}} \right) + \text{prod} \left(\frac{\partial J}{\partial s}, \frac{\partial s}{\partial \mathbf{W}^{(2)}} \right) = \frac{\partial J}{\partial \mathbf{o}} \mathbf{h}^T + \lambda \mathbf{W}^{(2)} \quad (11)$$

Para calcular el gradiente con respecto a $\mathbf{W}^{(1)}$ se necesita continuar con la propagación hacia atrás a lo largo de la capa de salida hasta la capa oculta, de esta forma el gradiente con respecto a la salida de la capa oculta $\partial J/\partial \mathbf{h}$ es:

$$\frac{\partial J}{\partial \mathbf{h}} = \text{prod} \left(\frac{\partial J}{\partial \mathbf{o}}, \frac{\partial \mathbf{o}}{\partial \mathbf{h}} \right) = \mathbf{W}^{(2)T} \frac{\partial J}{\partial \mathbf{o}} \quad (12)$$

Debido a que la función de activación ϕ se aplica por elementos, para calcular el gradiente $\partial J/\partial \mathbf{z}$ se requiere usar el operador \odot , como se muestra:

$$\frac{\partial J}{\partial \mathbf{z}} = \text{prod} \left(\frac{\partial J}{\partial \mathbf{h}}, \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \right) = \frac{\partial J}{\partial \mathbf{h}} \odot \phi'(\mathbf{z}) \quad (13)$$

Finalmente, se puede obtener el gradiente $\partial J/\partial \mathbf{W}^{(1)}$ de los parámetros más cercanos a la capa de entrada. De acuerdo con la regla de la cadena se tiene:

$$\frac{\partial J}{\partial \mathbf{W}^{(1)}} = \text{prod} \left(\frac{\partial J}{\partial \mathbf{z}}, \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \right) + \text{prod} \left(\frac{\partial J}{\partial s}, \frac{\partial s}{\partial \mathbf{W}^{(1)}} \right) = \frac{\partial J}{\partial \mathbf{z}} \mathbf{x}^T + \lambda \mathbf{W}^{(1)} \quad (14)$$

3.3. Algoritmos de optimización

Una red neuronal consta de muchos parámetros para modelar una solución al problema que están tratando, estos valores son conocidos como pesos. El trabajo de estos algoritmos de optimización es encontrar los mejores valores de los parámetros para reducir el error cometido por

la red, esto es posible con la ayuda de backpropagation y descenso del gradiente. En principio, los optimizadores actualizan estos parámetros equitativamente en base a una tasa de aprendizaje. Algunos de estos algoritmos se detallan a continuación [22].

3.3.1. Optimizador descenso del gradiente estocástico (SGD por sus siglas en inglés)

Funciona en base al descenso del gradiente que toma todo el conjunto de datos por cada iteración para optimizar los pesos, donde se evalúa el error del modelo y se calcula las derivadas parciales del error respecto a los parámetros. Esto genera un vector gradiente que indica la dirección de la pendiente hacia donde el error se incrementa y este se usa para moverse en la dirección contraria. De tal manera, iterativamente se va reduciendo el error del modelo. La diferencia principal del SGD con respecto al descenso del gradiente, es que selecciona una muestra aleatoria del conjunto de datos por iteración para alcanzar un mínimo del error. Esto genera que el costo de cálculo sea menor al descenso del gradiente, a pesar de requerir un número mayor de iteraciones para alcanzar los mínimos locales [23].

3.3.2. Optimizador Adam (Adaptive Moment Estimation)

Este algoritmo usa momentum y tasa de aprendizaje adaptativo para converger más rápido. El momentum ayuda a progresar más rápido en la red neuronal, donde puede ayudar a evitar que el optimizador se quede en un mínimo local en el error cometido por la red y evitar la oscilación excesiva.

En cuanto a la tasa de aprendizaje adaptativo, usa una propiedad de AdaGrad y otra de RMSProp, que son otros algoritmos de optimización. AdaGrad hace que la tasa de aprendizaje disminuya su valor con el tiempo y RMSProp cambia la tasa de aprendizaje a medida que se avanza en el análisis del error para cada época.

De esta forma Adam actualiza la tasa de aprendizaje para los pesos de la red individualmente y es considerado uno de los mejores optimizadores [24]. De esta forma, Adam tiene un tiempo de

cálculo más rápido, pero el SGD generaliza los datos de mejor manera a coste de una baja velocidad de cálculo. Por lo tanto, la elección del optimizador depende del volumen y tipo de datos [23].

3.4. Estrategias de regularización

El sobreajuste es uno de los principales problemas cuando se usan algoritmos de aprendizaje automático. Este consiste en que el modelo obtiene una alta exactitud en la etapa de entrenamiento, pero su capacidad de generalización es muy baja lo que provoca un bajo rendimiento cuando se valida con un conjunto de prueba. El objetivo principal es reducir la complejidad del modelo durante el entrenamiento para evitar el sobreajuste, algunas de estas técnicas se especifican a continuación.

3.4.1. Parámetro de regularización L2

La idea en el entrenamiento de los modelos es encontrar los pesos que minimicen la función de costo para conseguir mejores resultados en la clasificación, con este parámetro de regularización $L2$ puede lograrse este objetivo debido a que evita que los pesos aumenten demasiado, por el contrario hace que tiendan a cero al añadir el término de regularización $\Omega(\theta) = \frac{1}{2} \|\mathbf{w}\|_2^2$ a la función de costo [25]. Por otra parte, la disminución en los pesos mediante la regularización $L2$ minimiza el efecto de correlación entre las características de entrada y todo esto genera un modelo menos complejo evitando el sobreajuste consiguiendo una mejor generalización de este [26].

3.4.2. Early stopping

Lo que hace esta estrategia es almacenar una copia de los parámetros del modelo cada vez que el error en el conjunto de validación mejora. El entrenamiento del modelo termina cuando la pérdida en el conjunto de validación no ha mejorado durante un número de iteraciones previamente especificado. Esta estrategia permite reducir el tiempo de entrenamiento y evita que el modelo se sobreajuste [25].

3.4.3. Dropout

Esta estrategia funciona de tal forma que, en la etapa de entrenamiento con cierta probabilidad, algunas neuronas de la red neuronal son apagadas [27]. En la etapa de entrenamiento el objetivo principal es reducir la función de costo teniendo en cuenta todas las neuronas, de tal forma, en el sobreajuste es posible que una neurona pueda inferir en la corrección del error en otras neuronas y esto conlleva a unas coadaptaciones complejas, esta a su vez conduce al problema indeseado del sobreajuste sin alcanzar la generalización para el conjunto de validación. Por este motivo, al aplicar dropout se evita la coadaptación entre neuronas debido a que en cada época la presencia de una neurona tiene poca fiabilidad y esto ayuda en la generalización del modelo minimizando el problema del sobreajuste [28].

3.5. Redes neuronales convolucionales

Una red neuronal convolucional es una arquitectura de red para el aprendizaje profundo, generalmente son útiles para el reconocimiento de patrones en imágenes (2D), pero también funcionan para clasificar datos como audio, series temporales y datos de señales (1D) [17]. Típicamente, estas arquitecturas se componen de tres capas principales: capa convolucional, capa de agrupación y capa totalmente conectada [29].

3.5.1. Capas convolucionales

La operación matemática de convolución es realizada en esta capa y el objetivo de esto es extraer las características de la matriz (para las 2D-CNNs) de datos de entrada. La convolución usa un filtro, denominado en la literatura como kernel, de esta forma, en las primeras capas de convolución se extraen características de bajo nivel, que serán las entradas para las capas de convolución posteriores, repitiéndose este proceso de acuerdo con el número de capas definidas, al final se obtienen características de alto nivel. Si la entrada es una imagen, una característica de bajo nivel se asocia con la detección de patrones simples, como bordes y texturas, y las características de alto nivel se refiere a los patrones más complejos, como detección de objetos o rostros. De esta

manera, el modelo entrega un mapa de características que brinda una información completa sobre la matriz de datos ingresada [30].

La operación de convolución en su forma general, opera sobre dos funciones con argumento de valor real. Suponiendo que se tiene una función $x(t)$, donde x y t son valores reales y puede describir por ejemplo la posición de un objeto en el tiempo. Si se quisiera promediar dicha función se puede aplicar una función de ponderación $w(a)$ donde a es la época de una medición, al aplicar la ponderación en cada momento, se obtiene una función s que suaviza la función $x(t)$, en general es aplicable a muchos propósitos. La convolución se puede expresar de la siguiente manera:

$$s(t) = \int x(a) w(t - a) da; \quad s(t) = (x * w) \quad (15)$$

Normalmente en aplicaciones de aprendizaje automático la entrada y el kernel son matrices multidimensionales y se refiere a estos como tensores, de esta forma, la convolución opera sobre más de un eje al tiempo. Por ejemplo, cuando se trabaja con imágenes se tiene una entrada y kernel de dos dimensiones, y la convolución discreta para este caso es:

$$\mathbf{S}(i, j) = (\mathbf{I} * \mathbf{K})(i, j) = \sum_m \sum_n \mathbf{I}(m, n) \mathbf{K}(i - m, j - n) \quad (16)$$

Dónde \mathbf{I} es la matriz de información y \mathbf{K} es el kernel. Dado que la convolución es conmutativa, (16) se puede reescribir como aparece en (17).

$$\mathbf{S}(i, j) = (\mathbf{K} * \mathbf{I})(i, j) = \sum_m \sum_n \mathbf{I}(i - m, j - n) \mathbf{K}(m, n) \quad (17)$$

Generalmente se usa (17) en las librerías de aprendizaje automático, puesto que es más sencilla de implementar debido a la menor variación en el rango de valores válidos para m y n .

3.5.2. Capas de agrupación

Generalmente después de cada capa convolucional le sigue una capa de agrupación, el objetivo con esto es que realice una reducción en el tamaño del mapa de características convolucionado, con el fin de reducir costos computacionales. Básicamente resumen el mapa de características y para ello existen diversas operaciones de agrupación. Las más comunes son *Max Pooling*, *Average Pooling* y *Sum Pooling*. En *Max Pooling* toma el elemento más grande del mapa de características, en *Average Pooling* calcula el promedio de los elementos y *Sum Pooling* realiza la suma total de los elementos, todas las operaciones actúan sobre secciones predefinidas del mapa de características [30], estas se pueden apreciar en la **Fig. 2**.

Mapa de características	Max Pooling	Average Pooling	Sum Pooling																												
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>6</td><td>6</td><td>6</td><td>6</td></tr> <tr><td>4</td><td>5</td><td>5</td><td>4</td></tr> <tr style="border: 2px solid red;"><td>2</td><td>4</td><td>4</td><td>2</td></tr> <tr style="border: 2px solid red;"><td>2</td><td>4</td><td>4</td><td>2</td></tr> </table>	6	6	6	6	4	5	5	4	2	4	4	2	2	4	4	2	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>6</td><td>6</td></tr> <tr style="border: 2px solid red;"><td>4</td><td>4</td></tr> </table>	6	6	4	4	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>5.25</td><td>5.25</td></tr> <tr style="border: 2px solid red;"><td>3</td><td>3</td></tr> </table>	5.25	5.25	3	3	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>21</td><td>21</td></tr> <tr style="border: 2px solid red;"><td>12</td><td>12</td></tr> </table>	21	21	12	12
6	6	6	6																												
4	5	5	4																												
2	4	4	2																												
2	4	4	2																												
6	6																														
4	4																														
5.25	5.25																														
3	3																														
21	21																														
12	12																														

Fig. 2. Capas de agrupación comunes.

3.5.3. Capa totalmente conectada

Generalmente, se colocan antes de la capa de salida. Entonces, el mapa de características final se aplanan y se convierten en entrada de la capa totalmente conectada, que se denomina perceptrón multicapa. Este vector aplanado pasa por otras capas de este mismo tipo y posteriormente empieza el proceso de clasificación [30]. Esta capa permite reducir la dimensión del mapa de características aplanado para luego pasar a la última capa de la CNN, donde el número de neuronas es igual al número de clases a predecir, para determinar la pertenencia de una muestra a una clase, se hace uso de funciones de activación, siendo las más comunes la Softmax y la Sigmoide [31].

3.6. CNNs Unidimensionales

Las CNNs unidimensionales (1D-CNN) son similares a las tradicionales CNNs bidimensionales, la diferencia radica en que la entrada es un conjunto de secuencias temporales y no un conjunto de matrices. Esto permite que la complejidad de la arquitectura sea más baja y por lo tanto requiera menos muestras de entrenamiento para ser ajustada correctamente [32]. La **Fig. 3** muestra un ejemplo de una 1D-CNN.

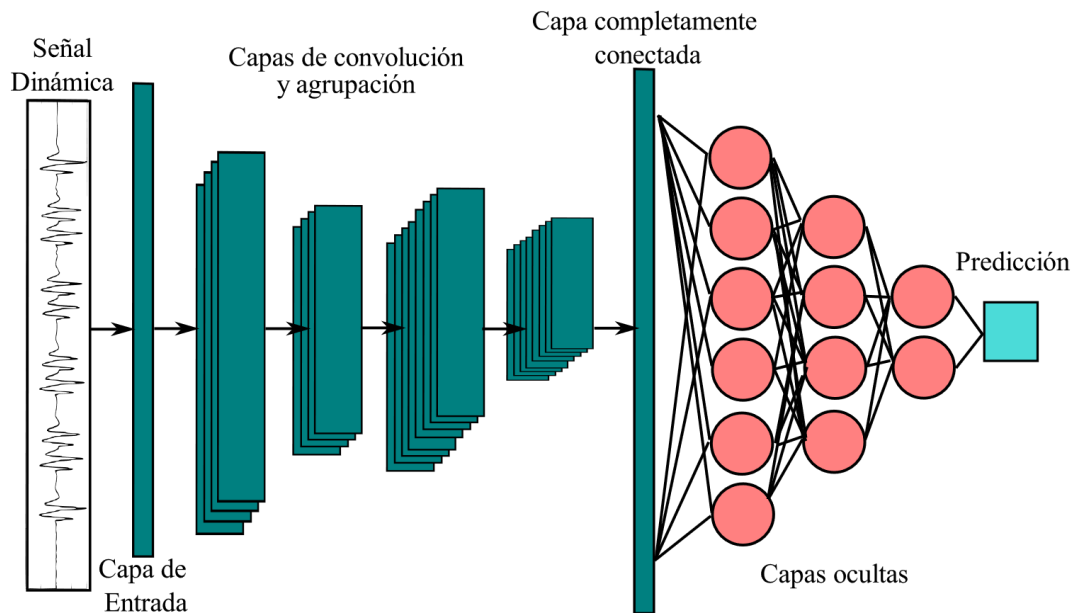


Fig. 3. Arquitectura de una 1D-CNN.

Los *kernels* de convolución siempre tiene el mismo ancho de la serie temporal, mientras que su longitud puede variar. El *kernel* entonces se mueve desde el comienzo hasta el final de la serie de tiempo, con un tamaño de paso definido. La salida de la capa convolucional es una nueva serie de tiempo sub-muestrada y filtrada. El número de las nuevas series de tiempo dependen del número de *kernels* de convolución definidos en la arquitectura. Adicionalmente, dependiendo de la longitud del *kernel* diferentes características de la serie temporal son extraídas en cada una de las nuevas series temporales. Finalmente se aplica una capa de *Max Pooling* similar a la aplicada en la 2D-CNN. La salida de la capa anterior es usada para alimentar una capa totalmente conectada, la cual realizará el proceso de clasificación [33].

El tamaño de paso definido con el que se desplaza el *kernel* se conoce como *stride*, dependiendo del tamaño de este parámetro se puede generar una mayor o menor cantidad de series de tiempo sub-muestradas y filtradas, debido a que realiza un solapamiento o no en la operación de convolución con la serie de tiempo ingresada. Finalmente, el *padding* es un parámetro que indica una cantidad de información adicional que se desea agregar a la serie temporal, normalmente valores de cero en los extremos, aumentando su longitud con la finalidad de que al realizarse la operación de convolución la salida sea de una dimensión mayor que al usar la serie de tiempo original [34].

Por otra parte, en estos casos se trabaja con tiempo discreto por lo que se define la convolución discreta de (15) que sería la aplicada en una 1D-CNN, se tiene:

$$\mathbf{s}(t) = (\mathbf{x} * \mathbf{w}) = \sum_{a=-\infty}^{\infty} \mathbf{x}(a)\mathbf{w}(t - a) \quad (18)$$

Donde \mathbf{x} sería la serie temporal de entrada, \mathbf{w} el kernel o filtro y la salida \mathbf{s} se le conoce como un mapa de características.

En las CNNs ocurre una reducción en la dimensionalidad de la entrada cada vez que pasa por una capa convolucional cuya salida se conoce como el mapa de características. La nueva dimensión se puede calcular con la siguiente fórmula, tanto para 2D-CNN como para 1D-CNN:

$$n_{out} = \left\lceil \frac{n_{in} + 2p - k}{s} \right\rceil + 1 \quad (19)$$

Donde n_{in} es el número de características de entrada, n_{out} el número de características de salida, p tamaño del padding en la convolución, k el tamaño del kernel de convolución y s el tamaño del stride de convolución.

En la **Fig. 4**, se muestra un ejemplo sobre el comportamiento en la reducción de la dimensionalidad para 1D-CNN teniendo como entrada una señal temporal de 6 segundos con una frecuencia de muestreo de 180 Hz lo que genera un vector de entrada de 1×1080 para la arquitectura.

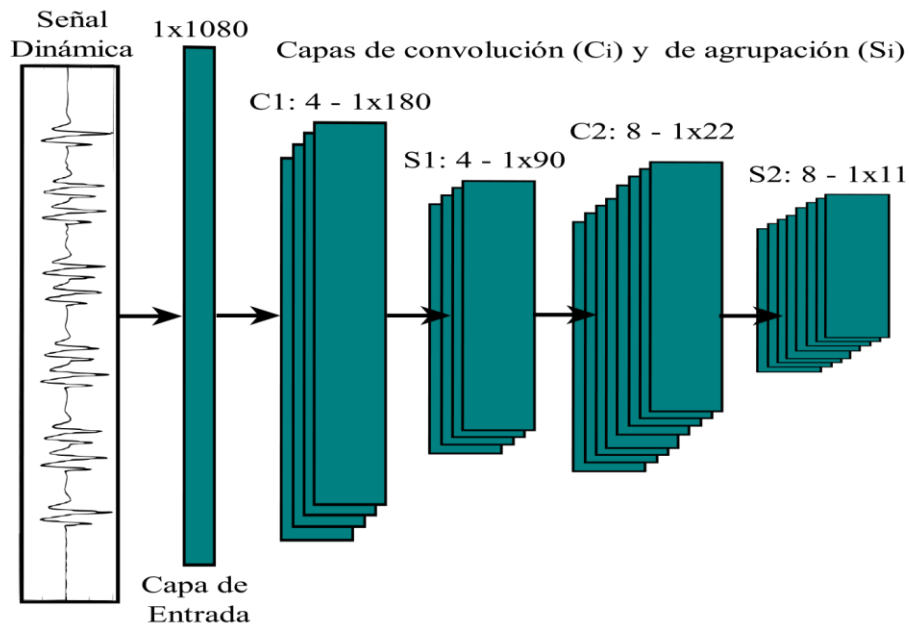


Fig. 4. Reducción de dimensionalidad en una señal dinámica en una 1D-CNN.

En la figura anterior, se aprecia que la entrada corresponde a una señal de 1×1080 , el tamaño de la capa de agrupación *Max Pooling* es de 2 (S1 y S2) y para la primera capa convolucional (C1) se tienen 4 *kernels* donde k y s es 6, para C2 se tienen 8 *kernels* donde k y s es 4. En ambas capas convolucionales p es 0. De esta forma aplicando (19) en la capa de entrada, se tiene:

$$n_{out} = \left\lceil \frac{1080 + 2(0) - 6}{6} \right\rceil + 1 = 180$$

Así entonces se tiene para C1: $4 - 1 \times 180$, donde 4 son el número de mapas de características generados con una dimensión de 1×180 . Es decir, se obtienen 4 series temporales de características que en comparación con el vector de entrada ha presentado una reducción en su tamaño de 6 veces, esto se debe a que el tamaño de k y s son iguales, donde no se genera un solapamiento entre la

información tomada por el *kernel* al realizar la operación de convolución. A manera de ejemplo, si en este caso se definiera el valor de s como la mitad del valor k entonces se generaría un solapamiento en la información capturada por el *kernel* y la reducción de tamaño sería de 3 veces. Aunque estas salidas son de menor resolución la duración es la misma en comparación con la señal de entrada, por otra parte, la frecuencia de muestreo cambia de 180 Hz a 30 Hz y esta disminución depende del tamaño del kernel.

Al aplicar *Max Pooling* sería $180/2 = 90$. Entonces, para S1: 4 - 1x90. En esta ocasión también ocurre un cambio en la frecuencia de muestreo cuya variación está determinada por el tamaño de la capa de agrupación es decir pasa de 30 Hz a 15 Hz. Debido a este submuestreo es que los vectores de características reducen su dimensión a 1x90.

Posteriormente, aplicando (19) y el mismo procedimiento explicado para C1 y S1, se tiene:

$$n_{out} = \left[\frac{90 + 2(0) - 4}{4} \right] + 1 = 22,5$$

Se tiene para C2: 8 - 1x22 y para S1: 8 - 1x11, donde la frecuencia de muestreo final se modifica de 15 Hz a 3.75 Hz por el submuestreo realizado en C2 y de 3.75 Hz a 1.875 Hz debido a la reducción de resolución realizada en S2. Es así como se tienen 8 vectores de características finales de tamaño 1x11 para la serie temporal ingresada de 1x1080. De esta forma ocurre la reducción de dimensionalidad y cambios de la frecuencia de muestreo en las capas convolucionales y de agrupación en una 1D-CNN.

Finalmente, en estudios recientes la mayoría de las aplicaciones con 1D-CNN han utilizado configuraciones compactas (con 1 o 2 capas convolucionales) donde generalmente las arquitecturas se componen de 10.000 parámetros, mientras que en las arquitecturas donde se consideran 2D-CNN tienen más de un millón de parámetros [35].

3.7. Estrategias de validación

Es una práctica común al realizar experimentos en los modelos dividir el conjunto de datos en dos: conjunto de entrenamiento y conjunto de prueba, donde el conjunto de entrenamiento se usa para optimizar los hiperparámetros del modelo y el conjunto de prueba se utiliza para evaluar el desempeño del modelo [36]. Estos conjuntos de datos se usan en una técnica general para detectar el sobreajuste conocida como validación cruzada con la finalidad de validar si el modelo generaliza correctamente. Existen diversas técnicas de validación como Hold-out, K-fold y Leave-one-out [37]. En este trabajo se considera específicamente la técnica de validación cruzada K-fold.

3.7.1. Validación cruzada K-fold

Esta técnica se utiliza cuando no se tienen suficientes datos disponibles, en esta estrategia de validación cruzada se toma del conjunto de datos en k folds (pliegues o subconjuntos), donde k es un número entero. Los $k-1$ folds son tomados para realizar el entrenamiento del modelo y, posteriormente, el fold reservado se usa para evaluar el modelo. Este proceso se realiza k veces, donde el fold reservado para evaluar, a su vez excluido en el entrenamiento, es distinto en cada iteración. Típicamente, el rendimiento del modelo se calcula usando la media aritmética de las métricas de desempeño usadas en las k iteraciones realizadas [38]. En la **Fig. 5** se muestra cómo funciona esta estrategia para un $k=5$.

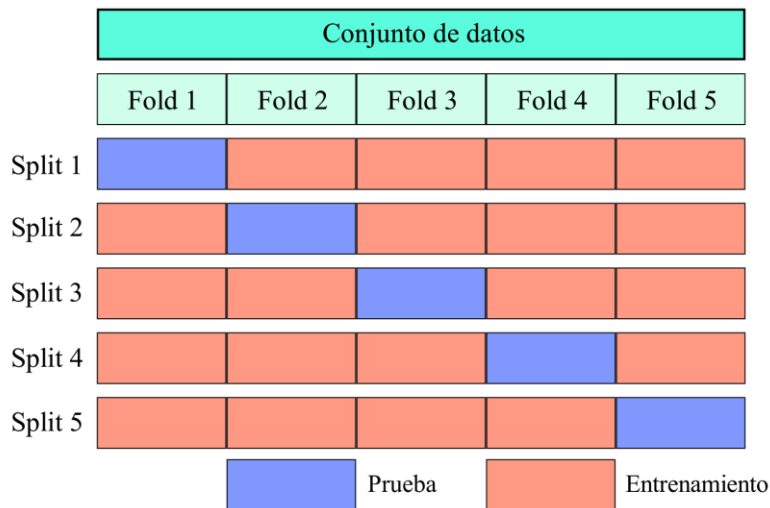


Fig. 5. validación cruzada k-fold con $k=5$

3.8. Métricas de desempeño

Evaluar el desempeño de un clasificador de aprendizaje automático es importante para validar la efectividad en la discriminación entre las clases, algunas de las métricas de desempeño más importantes son: matriz de confusión, sensibilidad, especificidad, exactitud, precisión, F1-score, curva ROC y área bajo la curva (AUC por sus siglas en inglés).

3.8.1. Matriz de confusión

La matriz de confusión es utilizada generalmente para evaluar el rendimiento de un modelo de clasificación, siendo una matriz cuadrada cuyo tamaño depende del número de clases. Una matriz de confusión de dos clases identifica una como la clase *positiva* (ej. Pacientes con EP) y la otra como *negativa* (ej. Sujetos de CS). De esta forma, las filas de la matriz corresponden a los etiquetas predichos y las columnas indican los etiquetas reales (o viceversa) [39], en la **Fig. 6** se muestra una matriz de confusión. Los datos de cada celda de la matriz se interpretan de la siguiente forma:

- True Positive (TP): Muestras que el modelo predice correctamente de la clase positiva.
- True Negative (TN): Muestras que el modelo predice correctamente de la clase negativa.
- False positive (FP): Muestras que el modelo predice incorrectamente de la clase positiva.
- False negative (FN): Muestras que el modelo predice incorrectamente de la clase negativa.

Reales	TN	FP
	FN	TP
	Predichas	

Fig. 6. Ejemplo de matriz de confusión.

De la matriz de confusión se pueden extraer otras métricas para evaluar el desempeño del modelo como:

Exactitud: es la proporción de predicciones que el modelo clasifica correctamente del conjunto total de datos.

$$Exc = \frac{TP + TN}{TP + FP + TN + FN} \quad (20)$$

Precisión: También es conocida como el valor predictivo positivo, indica que tan bueno es el modelo para reconocer las muestras de la clase positiva.

$$Pre = \frac{TP}{TP + FP} \quad (21)$$

Especificidad: también conocida como la tasa de verdaderos negativos (TNR por sus siglas en inglés), mide la proporción de negativos actuales que se clasificaron correctamente.

$$Esp = \frac{TN}{TN + FP} \quad (22)$$

Sensibilidad: se conoce como *Recall* o la tasa de verdaderos positivos (TPR por sus siglas en inglés), es la proporción de positivos actuales clasificados correctamente.

$$Sen = \frac{TP}{TP + FN} \quad (23)$$

F1-Score: Es el promedio ponderado entre la sensibilidad y la precisión.

$$F1 - Score = \frac{2 * Pre * Sen}{Pre + Sen} \quad (24)$$

3.8.2. Curva ROC

La curva ROC (Receiver Operating Characteristic) se construye al mover el umbral de izquierda a derecha en las distribuciones de cada clase, las cuales son generadas a partir de los scores del clasificador. A la derecha de la **Fig. 7** se muestran las distribuciones de la clase positiva y negativa para la curva ROC mostrada a la izquierda en la misma figura.

3.8.3. Área bajo la curva ROC (AUC)

El AUC es el área bajo la curva ROC y representa el grado o medida de separabilidad entre clases, es decir, indica la capacidad del modelo de distinguir entre clases. Cuando el AUC es 0.5 significa que el modelo no tiene la capacidad de discriminar entre clases, mientras que si el AUC es 1 indica una situación ideal donde el modelo es perfectamente capaz de distinguir entre clases [40]. La **Fig. 7** muestra la curva ROC con su correspondiente valor AUC.

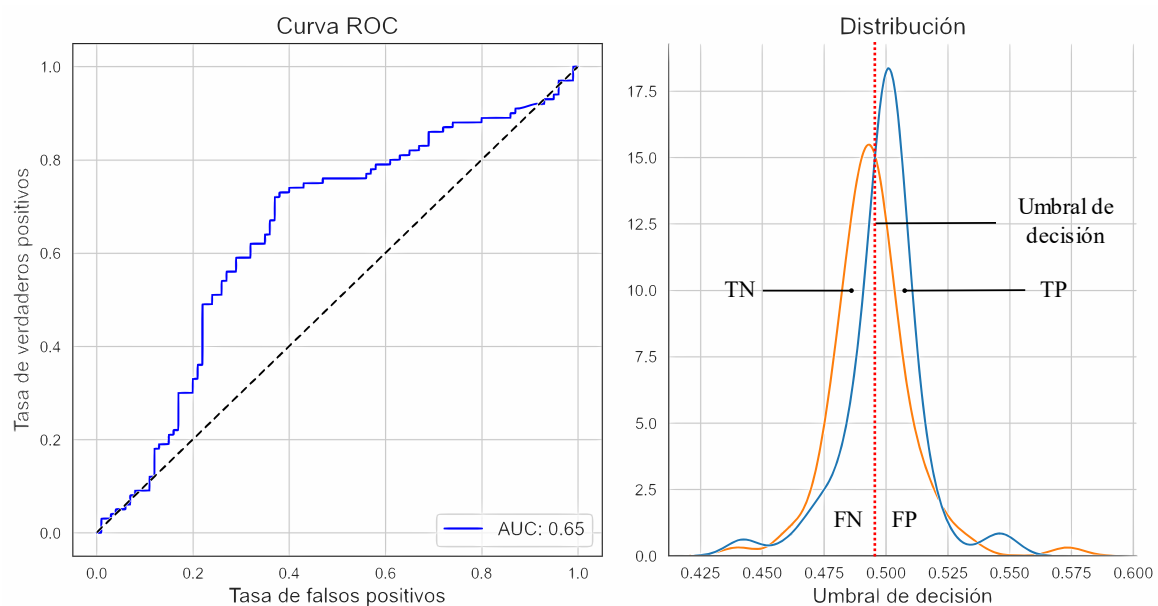


Fig. 7. Curva ROC y AUC.

4. BASE DE DATOS

La base de datos está compuesta por 75 sujetos con EP y 55 sujetos de CS. Los participantes desarrollaron diferentes tareas de escritura usando la tableta digitalizadora Wacom Cintiq 13 HD, la cual tiene una frecuencia de muestreo de 180 Hz. Esta tableta captura seis señales diferentes en forma de series temporales: coordenadas cartesianas (x, y, z), ángulo azimut, ángulo de altitud y presión. Específicamente, en este trabajo se usó la tarea de espiral de Arquímedes, la cual ha sido usada exitosamente para evaluar el deterioro grafomotor de los pacientes con EP [6]. También se cuenta con la evaluación de un neurólogo experto según la Escala Unificada para la Evaluación de la Enfermedad de Parkinson (UPDRS). La **TABLA I** resume la información demográfica y clínica de los sujetos.

TABLA I. Información demográfica y clínica de los sujetos. μ : media, σ : desviación estándar

	Pacientes con EP		Sujetos de CS	
	Hombres	Mujeres	Hombres	Mujeres
Cantidad de sujetos	35	40	33	22
Edad ($\mu \pm \sigma$)	67.6 \pm 10.6	60.8 \pm 12.1	67.0 \pm 10.4	58.6 \pm 10.5
Rango de edad	41 – 90	29 – 85	49 – 85	43 – 83
UPDRS-III ($\mu \pm \sigma$)	35.9 \pm 20.2	37.8 \pm 16.7	---	---
Rango de UPDRS-III	9 – 106	12 – 82	---	---

5. METODOLOGÍA

La metodología general que se abordará en este trabajo se resume en la **Fig. 8**. Primero se toma la tarea de escritura de espiral de Arquímedes para sujetos con EP y CS. Este proceso de escritura online se refiere al hecho de realizar tareas de escritura a mano en dispositivos electrónicos que capturan información en forma de series temporales. En este trabajo la tableta digitalizadora usada para la escritura online genera seis señales dinámicas diferentes: coordenadas cartesianas (x, y, z), ángulo azimut, ángulo de altitud y presión. Adicionalmente, se realiza un proceso de segmentación en la serie temporal tomando una ventana de análisis para varias resoluciones temporales, el motivo para realizar esta segmentación es con el fin de generar subsecuencias más cortas de las series temporales, donde al tener una menor resolución temporal será mayor la

cantidad de segmentos creados, pero su dimensión es reducida. Por otro lado, al tener una ventana de análisis de mayor tamaño se generarían pocas muestras, pero su dimensión y la información que abarca cada segmento sería mayor. Tener más muestras para ajustar la arquitectura puede ser beneficioso puesto que las redes neuronales convolucionales generalmente necesitan gran cantidad de ellas para que el modelo aprenda las características necesarias y así realizar una discriminación entre clases con mejor desempeño, sin embargo, una dimensión mayor en los segmentos podría favorecer la exactitud en la clasificación de EP vs CS al contener más información.

Posteriormente, estos segmentos son tomados como entrada a una arquitectura 1D-CNN para dos modelos diferentes: modelo monocanal y multicanal, donde finalmente en la etapa de clasificación, se decide si el segmento pertenece a un sujeto con EP o por el contrario a un CS. Los detalles de los modelos monocanal y multicanal se encuentran en las siguientes secciones.

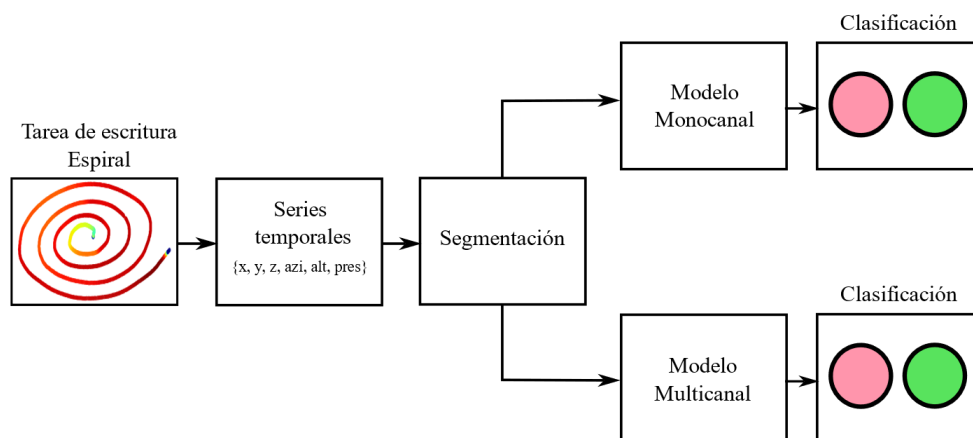


Fig. 8. Metodología general.

Por otra parte, el conjunto de datos se encuentra desbalanceado porque se tienen 55 sujetos CS y 75 sujetos EP, entonces, se toman de manera aleatoria 55 sujetos EP de los 75 disponibles logrando así balancear el conjunto de datos. Además, en el proceso de segmentación de las series temporales es necesario tener presente que cada sujeto realiza la tarea de escritura de acuerdo con sus capacidades y limitaciones, por tanto, la duración de la prueba es distinta en todos los sujetos.

La tasa de muestreo de la tablet digitalizadora es de 180 Hz, entonces se tiene que por cada segundo se tienen 180 puntos o datos, donde cada punto corresponde 5,56 ms de información sobre

una tarea de escritura online. De esta forma, si el tamaño del kernel es de 8 entonces se analiza 44,44 ms de información, es así como al modificar el tamaño del kernel se tienen distintas ventanas de análisis sobre los segmentos de la serie temporal ingresada en la primera capa convolucional, entonces se configura este tamaño de tal forma que permita analizar ventanas de 44,44 ms, 88,89 ms y 177,78 ms de información.

Cómo se explicó anteriormente la idea es generar una mayor cantidad de muestras para el entrenamiento y prueba de las diferentes arquitecturas 1D-CNN al seleccionar distintas resoluciones temporales para el proceso de segmentación, teniendo presente que en algunos trabajos como en Taleb et al. afirman que la limitada disponibilidad de datos para la clasificación de pacientes con EP vs CS es desafiante al usar aprendizaje profundo [12]. Por otra parte, la distribución de la duración en la tarea de escritura espiral de Arquímedes para todos los sujetos analizados en este trabajo se muestra en la **Fig. 9**, donde la mayor cantidad de información se concentra alrededor de los 12.1 segundos siendo este el promedio del conjunto de datos. Teniendo en cuenta todos estos aspectos descritos, los tamaños de las resoluciones temporales establecidas en este trabajo para el proceso de segmentación son de 0.5s, 1s, 2s, 4s, 6s y 12s.

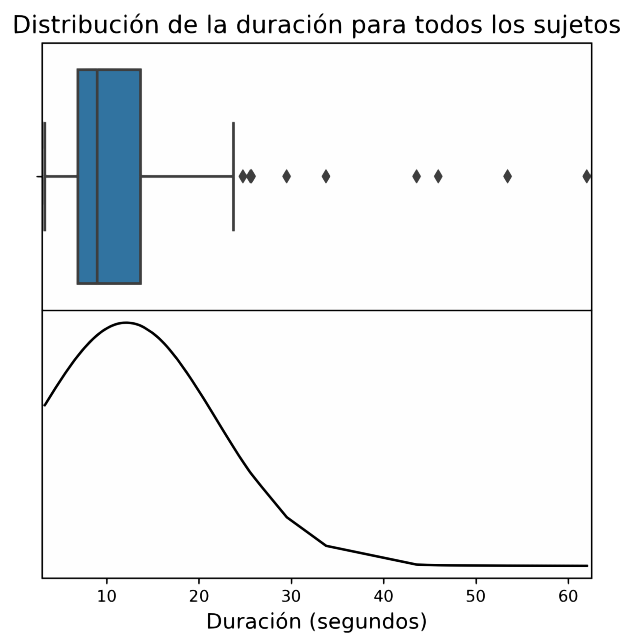


Fig. 9. Distribución de la duración en la prueba para todos los sujetos.

5.1. Experimentos

Un total de cuatro experimentos serán realizados en este trabajo que buscan evaluar el desempeño que tienen arquitecturas 1D-CNN con enfoques monocanal y multicanal para analizar el deterioro en habilidades grafomotoras de pacientes con EP, donde al final se comparan los enfoques propuestos para validar si es mejor un modelo monocanal o multicanal. En ambos modelos se realiza la clasificación por segmentos, pero adicionalmente a esto se realiza una clasificación de EP vs CS por sujeto, esto debido a que al final de la arquitectura 1D-CNN, es decir en la capa completamente conectada, se encuentra una neurona con una función de activación Sigmoide que entrega una probabilidad (score) de pertenencia a una clase, si es mayor que 0.5 el sujeto tiene la EP o de lo contrario es un CS, de esta forma se toman todos los segmentos pertenecientes a cada sujeto y se calcula el score promedio de estos, con la misma regla de clasificación anterior se determina si el sujeto tiene la EP o es un CS, realizando así una clasificación por sujeto. Las siguientes secciones describen con detalle los cuatro experimentos realizados.

5.1.1. Optimización de arquitectura monocanal.

Para el modelo monocanal se tienen diversos parámetros que se requieren optimizar estos son el tamaño de la resolución temporal de la ventana de análisis para las señales dinámicas siendo de {0.5s, 1s, 2s, 4s, 6s, 12s}, la tasa de aprendizaje en {1e-3, 1e-4}, el número de kernel inicial {8, 16, 32, 64}, el número de capas totalmente conectadas {8, 16, 32, 64, 128, 256, 512, 1024}, estos dos últimos parámetros corresponden a valores de potencias de 2 que son generalmente usados en las CNNs, del mismo modo, se tiene el solapamiento de las muestras generadas para de cada resolución temporal (La idea es generar el doble de muestras cuando se aplica) puede ser {False, True}, dropout {0, 0.2, 0.4, 0.6, 0.8} y finalmente el tamaño del kernel y stride donde adicionalmente se modifica el stride para realizar un solapamiento de la mitad en las ventanas de análisis definidas por el tamaño del kernel, entonces, los tamaños del kernel y stride son {(8,4), (16,8), (32,16)} para la primera capa convolucional, donde (kernel, stride), y la segunda capa convolucional tiene el tamaño del kernel y stride fijo en {(4,2)}.

La resolución temporal máxima de la ventana de análisis para las señales dinámicas es de 12s y con este valor se generan aproximadamente 160 segmentos, que serán el menor número de muestras generadas de las distintas resoluciones temporales definidas, debido a esto se fija el tamaño del batch en 32; en dónde se garantiza que el número de muestras por cada clase está balanceado para cada batch.

La metodología y diseño de la arquitectura 1D-CNN para el modelo monocanal se detalla en la **Fig. 10**, donde las señales se toman de forma independiente para ser ingresadas a una arquitectura 1D-CNN diferente para cada señal y que finalmente entregará una clasificación por segmento y sujeto. En este proceso de optimización de las arquitecturas no se aplica validación cruzada k-fold debido al elevado costo computacional para este proceso. Se realiza una división estratificada del conjunto de datos con 80% para entrenamiento y 20% para prueba.

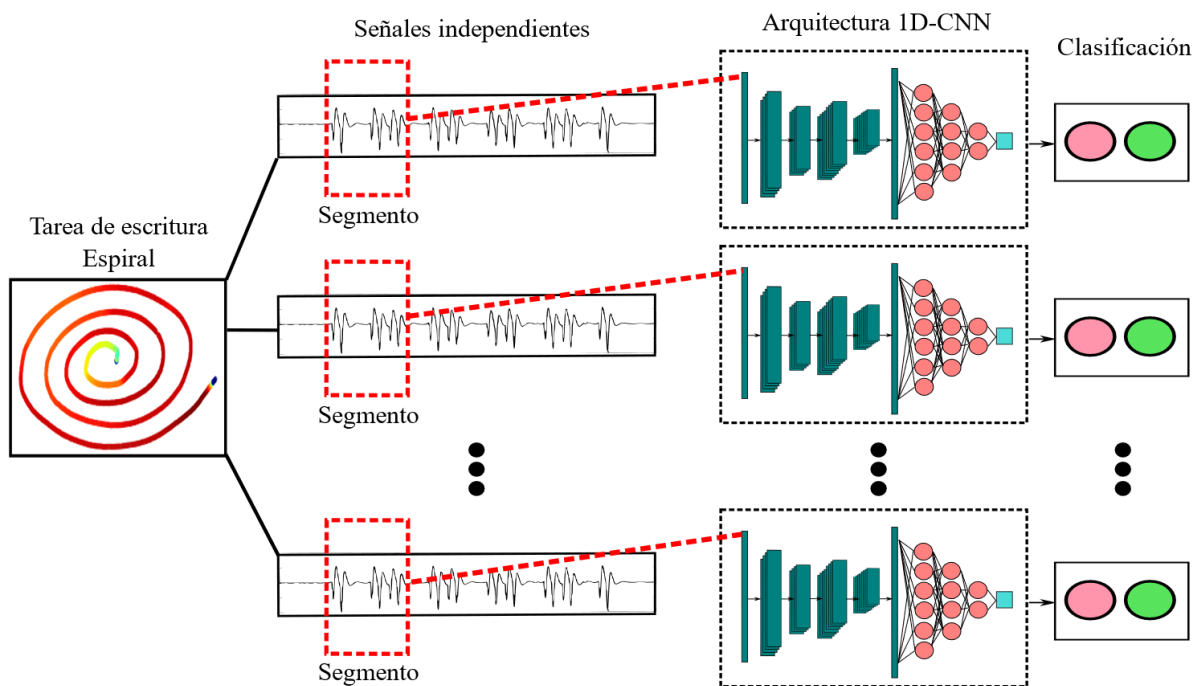


Fig. 10. Metodología para el modelo monocanal.

5.1.2. Validación cruzada del monocanal

Luego de obtener los mejores parámetros para los modelos monocanal, se procede con los experimentos de clasificación de pacientes con EP, en donde se usa la estrategia de validación cruzada k-fold de 5 folds para verificar la generalización de los modelos encontrados en los experimentos definidos en la sección **5.1.1**, se analizarán los resultados obtenidos de acuerdo con las métricas de desempeño usadas para la clasificación por segmentos y clasificación por sujeto.

5.1.3. Optimización de arquitectura multicanal

La optimización de los parámetros en este modelo multicanal se simplifica puesto que se toma como referencia los mejores parámetros encontrados en la optimización del modelo monocanal, por tanto, los parámetros en ambos modelos son los mismos que se definieron en la sección **5.1.1**, con la diferencia de que algunos parámetros pueden limitarse a un menor rango de valores o fijarse solo a uno, la modificación de los parámetros se realizará de acuerdo a los que presenten una mayor ocurrencia en las seis modelos monocanal.

Con respecto a la metodología para este modelo multicanal, se difiere del monocanal en que más de dos o todas las señales dinámicas generadas por el proceso de escritura online son agrupadas para aplicar el proceso de segmentación de acuerdo con las diferentes resoluciones temporales para las ventanas de análisis que fueron definidas, donde finalmente se toman como entrada al modelo multicanal generando así una clasificación de pacientes con EP vs CS, por segmentos y por sujeto, esto se muestra en la **Fig. 11**. En este experimento no se aplica validación cruzada y la división de los datos es igual al mencionado para el modelo monocanal.

5.1.4. Validación cruzada del multicanal

De forma similar a la metodología usada en la sección **5.1.2** del modelo monocanal, los mejores parámetros para el multicanal se toman para desarrollar los experimentos de clasificación de pacientes con EP por segmentos y sujeto, inicialmente se tomarán todas las señales para validar la generalización de este modelo mediante la validación cruzada y adicionalmente se probará un

modelo multicanal agrupando las señales dinámicas que entreguen un mejor desempeño en la validación cruzada del modelo monocanal esto con el fin de validar si la exactitud mejora en comparación con el modelo multicanal que agrupa todas las señales. Finalmente, se realizará la comparación de ambos enfoques basados en la exactitud obtenida.

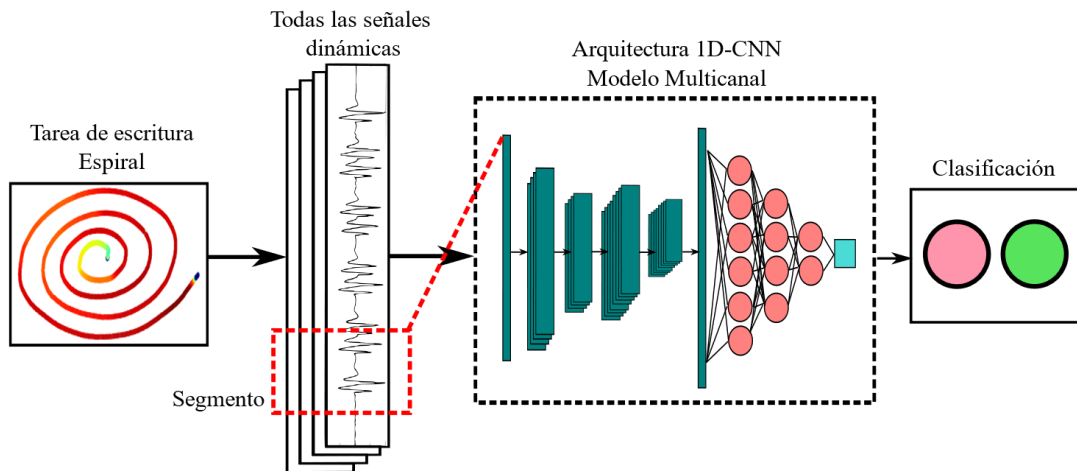


Fig. 11. Metodología para el modelo multicanal.

6. RESULTADOS

6.1. Optimización de la arquitectura monocanal

En la **TABLA II** se encuentran los mejores parámetros para las distintas arquitecturas 1D-CNN de los modelos monocanal correspondientes a las distintas señales dinámicas. Estos valores son usados en los experimentos de clasificación con la estrategia de validación cruzada. Por otra parte, la **TABLA III** muestra el desempeño de las distintas arquitecturas 1D-CNN en la clasificación por segmentos, el mejor resultado corresponde a la señal de altitud.

La **TABLA IV** muestra el desempeño de las mismas arquitecturas en la clasificación por sujeto, donde el mejor resultado se mantiene para la misma señal. En la **Fig. 12** se muestra a la izquierda la curva ROC obtenida para el mejor resultado en la optimización de la arquitectura para el modelo monocanal, siendo la señal de altitud la que demuestra mayor capacidad para clasificar pacientes con EP de los sujetos de CS, mostrando un área bajo la curva ROC de 0.87. A la derecha

de la misma figura se encuentra la distribución de los scores obtenidos en donde se evidencia cierto grado de separabilidad entre las clases generando exactitud de hasta un 70%.

TABLA II. Resultados de los mejores parámetros para la arquitectura 1D-CNN con el modelo monocanal. **Res.**: Resolución temporal, **KS**: Kernel y Stride, **Solp**: Solapamiento, **LR**: Tasa de aprendizaje, **Filt**: número de filtros iniciales, **FC**: capas completamente conectadas y **Dpo**: Dropout

Modelo monocanal – Mejores parámetros							
Señales	Res. Temp	KS	Solp	LR	Filt	FC	Dpo
X	6 s	(16,8)	False	1e-4	8	256	0.6
Y	6 s	(8,4)	False	1e-4	64	128	0.6
Z	12 s	(32,16)	True	1e-4	8	128	0.6
Altitud	6 s	(32,16)	False	1e-3	8	8	0.4
Presión	12 s	(32,16)	False	1e-4	16	256	0.4
Azimut	6 s	(16,8)	False	1e-3	16	256	0.6

TABLA III. Métricas de desempeño por segmentos para los mejores parámetros del modelo monocanal

Modelo monocanal – Segmentos				
Señales	Exactitud (%)	Sensibilidad (%)	Especificidad (%)	F1 score (%)
X	75.0	80.0	70.0	76.2
Y	70.0	75.0	65.0	71.4
Z	69.4	72.2	66.7	70.3
Altitud	77.5	65.0	90.0	74.3
Presión	73.1	61.5	84.6	69.6
Azimut	67.5	70.0	65.0	68.3

TABLA IV. Métricas de desempeño por sujeto para los mejores parámetros del modelo monocanal

Modelo monocanal – Sujeto				
Señales	Exactitud (%)	Sensibilidad (%)	Especificidad (%)	F1 score (%)
X	81.8	81.8	81.8	81.8
Y	77.3	72.7	81.8	76.2
Z	77.3	81.8	72.7	78.2
Altitud	86.4	81.8	90.9	85.7
Presión	72.7	63.6	81.8	70.0
Azimut	68.2	54.5	81.8	63.2

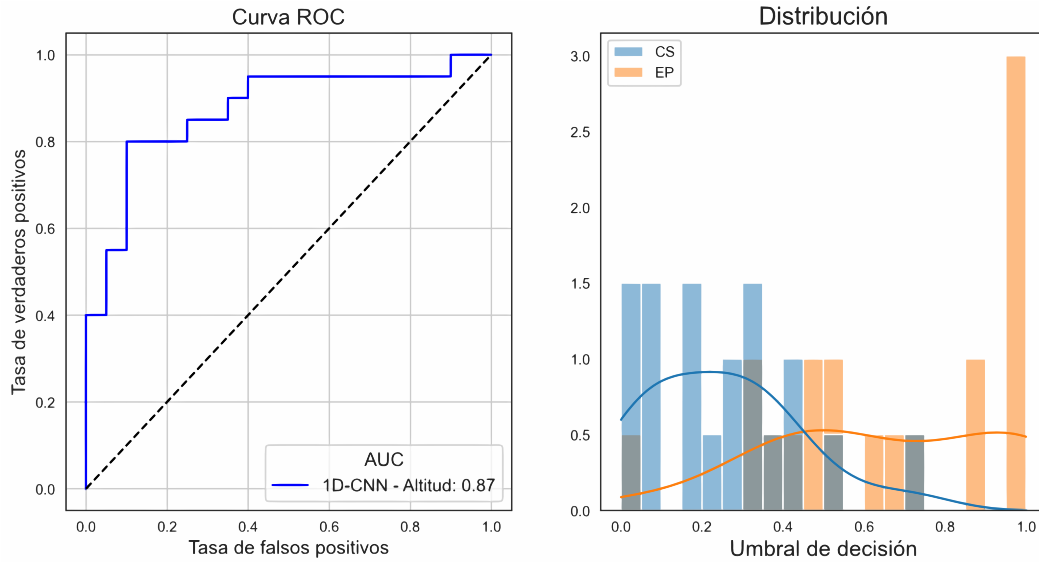


Fig. 12. Curva ROC y distribución de scores para el mejor resultado del modelo monocanal

Por otra parte, en la **Fig. 13** se muestran las curvas ROC para todas las señales analizadas con el modelo monocanal, además, en el mejor resultado del enfoque monocanal se tiene que la capacidad de reconocimiento del modelo presenta un sesgo hacia la clase EP, lo que se corrobora con los valores de especificidad y sensibilidad presentados en la **TABLA III**.

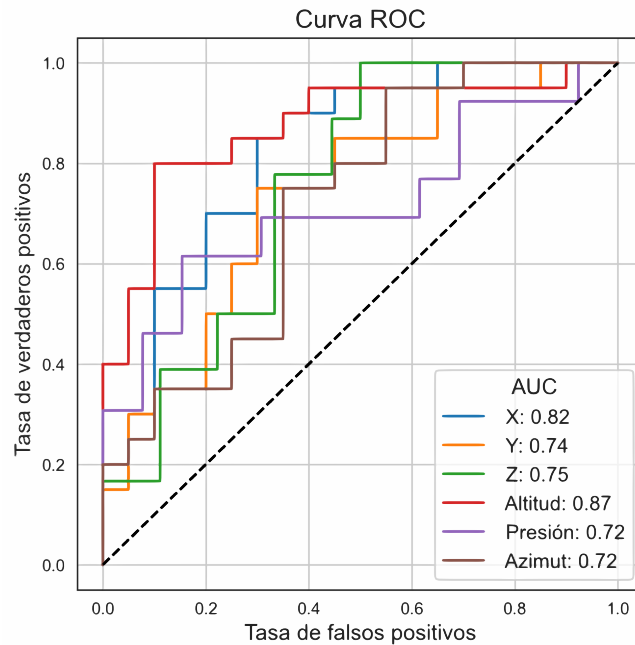


Fig. 13. Curvas ROC para los modelos monocanal.

6.1.1. Análisis del desempeño en las distintas resoluciones temporales

En la **Fig. 14** se muestran los mejores desempeños obtenidos para las distintas resoluciones temporales de las ventanas de análisis en los modelos monocanal, a la derecha de la figura se encuentran las mejores exactitudes de las seis series temporales para todos los tamaños de ventanas definidos y a la izquierda el promedio de las exactitudes para todas las señales. En 6s se encuentra el codo de saturación para obtener las mejores exactitudes en los modelos, sin embargo, la diferencia entre las exactitudes promedio de 6s y 12s no presentan una diferencia muy pronunciada y esto se corrobora con los mejores parámetros obtenidos en la **TABLA II**.

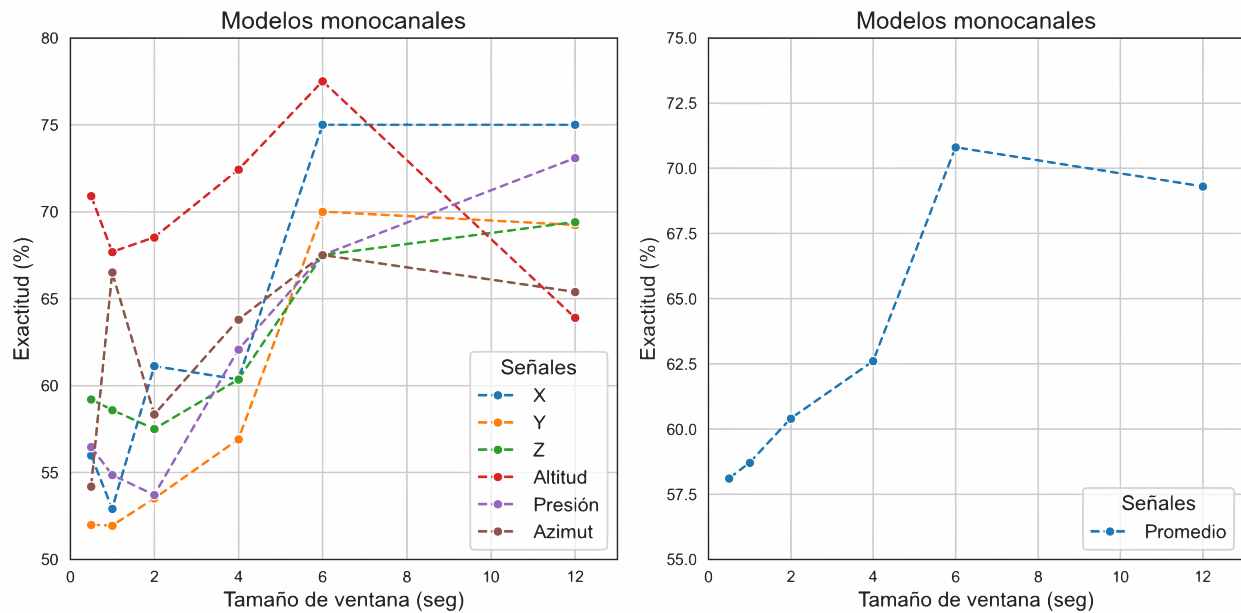


Fig. 14. Exactitudes obtenidas al varias la resolución temporal en los modelos monocanal

6.2. Validación cruzada del modelo monocanal

En la **TABLA V**, se presentan los resultados de la clasificación de pacientes con EP vs CS usando la estrategia de validación cruzada k-fold del modelo monocanal por segmentos, en donde la mayor exactitud se consigue para la señal de Altitud con un 61.6%. De igual manera, en la **TABLA VI**, se presentan los resultados por sujeto en donde igualmente se mantiene la señal de Altitud con el mejor desempeño con una exactitud del 64.0%.

TABLA V. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo monocanal con los segmentos. μ : media, σ : desviación estándar

Modelo monocanal – Segmentos				
Señales	Exactitud ($\mu \pm \sigma$)%	Sensibilidad ($\mu \pm \sigma$)%	Especificidad ($\mu \pm \sigma$)%	F1 score ($\mu \pm \sigma$)%
X	56.7±6.5	55.3±15.7	58.6±20.4	55.2 ± 8.8
Y	52.8±9.5	52.8±24.9	52.8 ± 8.7	50.1±17.9
Z	57.1±5.0	57.5±19.7	56.7±18.3	55.4±12.5
Altitud	61.6±7.8	64.6 ± 9.2	58.6±19.0	61.1±12.9
Presión	51.2±7.7	44.9±16.6	57.6±22.6	46.5±10.2
Azimut	52.7±9.3	60.5 ± 7.5	44.9±25.2	56.4 ± 2.6

TABLA VI. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo monocanal por sujeto. μ : media, σ : desviación estándar

Modelo monocanal – Sujeto				
Señales	Exactitud ($\mu \pm \sigma$)%	Sensibilidad ($\mu \pm \sigma$)%	Especificidad ($\mu \pm \sigma$)%	F1 score ($\mu \pm \sigma$)%
X	58.7±11.4	58.2±17.8	59.5±19.3	58.19±11.7
Y	48.5 ± 7.9	52.9±28.7	43.6±15.6	46.9±19.0
Z	56.4±10.6	52.7±24.0	60.0±20.4	51.8±20.1
Altitud	64.0 ± 8.4	60.9±20.7	66.4±13.4	61.4±11.3
Presión	50.5 ± 7.7	40.9±16.3	60.0±19.6	43.4±11.7
Azimut	58.8±15.8	70.2±13.8	47.3±26.6	63.2±10.8

6.3. Optimización de la arquitectura multicanal

Para este experimento se agrupan todas las señales en un modelo multicanal, la **TABLA VII** muestra el mejor desempeño para la clasificación por segmentos y sujeto. Los mejores parámetros fueron: kernel y stride de $\{(32,16)\}$, solapamiento $\{\text{False}\}$, tasa de aprendizaje $\{1e-4\}$, el número de kernel inicial $\{8\}$, capa completamente conectada $\{8\}$ y el dropout de $\{0.4\}$.

TABLA VII. Métricas de desempeño en la clasificación por segmentos y por sujeto en el modelo multicanal tomando todas las señales dinámicas

Modelo multicanal – Todas las señales				
Clasificación	Exactitud (%)	Sensibilidad (%)	Especificidad (%)	F1 score (%)
Segmentos	73.1	69.2	76.9	72.0
Sujeto	77.3	72.7	81.8	76.2

En la **Fig. 15** se muestra la curva ROC obtenida para el mejor resultado en la optimización de la arquitectura del modelo multicanal, con un valor AUC de 0.78, además en la gráfica de distribución de los scores se presenta una separabilidad entre las clases que permite obtener un porcentaje de exactitud de 73.1% en la clasificación, en general, en esta figura se aprecia que al agrupar todas las señales se podría generar buen desempeño en la clasificación de pacientes con EP de los sujetos de CS.

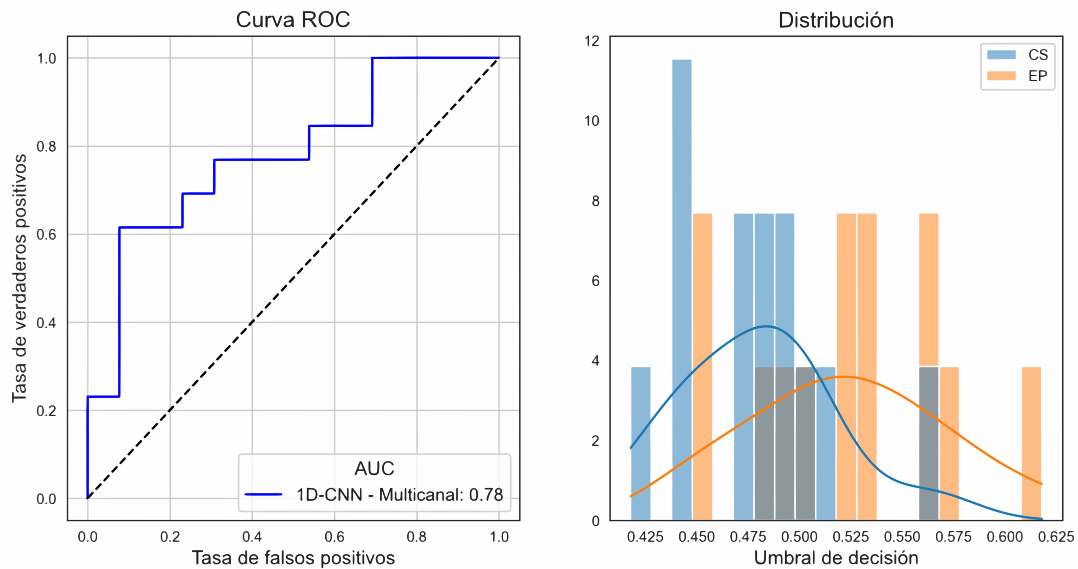


Fig. 15. Curva ROC y distribución de scores para el mejor modelo multicanal.

6.4. Validación cruzada del multicanal

En la **TABLA VIII** se presentan los resultados de la clasificación de pacientes con EP vs CS usando la estrategia de validación cruzada k-fold del modelo multicanal por segmentos y sujeto tomando todas las señales, en donde se obtiene una mayor exactitud en la clasificación por sujeto.

TABLA VIII. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo multicanal tomando todas las señales. μ : media, σ : desviación estándar

Modelo multicanal – Todas las señales				
Clasificación	Exactitud ($\mu \pm \sigma$)%	Sensibilidad ($\mu \pm \sigma$)%	Especificidad ($\mu \pm \sigma$)%	F1 score ($\mu \pm \sigma$)%
Segmentos	57.1±5.6	57.3±32.7	56.9±30.3	51.4±20.6
Sujeto	59.7±4.8	61.1±30.6	58.2±30.2	55.0±19.6

Debido a que en la **TABLA V** los tres mejores resultados en exactitud corresponden a las señales X, Z y Altitud, se realiza un experimento adicional con la finalidad de comprobar si al tomar estas tres señales se logra mejorar la exactitud en un enfoque multicanal, de esta manera, los resultados conseguidos para este experimento adicional se muestran en la **TABLA IX**, donde las exactitudes aumentan en comparación con las logradas al tomar todas las señales. La mejor exactitud se consigue en la clasificación por sujeto al agrupar las tres señales descritas, siendo este resultado igual al obtenido en la **TABLA VI**, sin embargo, para el modelo monocanal la sensibilidad y especificidad tienen menor variación entre sí lo que indica una mejor discriminación entre las clases sin sesgarse en alguna de las dos.

TABLA IX. Métricas de desempeño para la clasificación de pacientes con EP vs CS usando validación cruzada en el modelo multicanal tomando las señales X, Z y Altitud. μ : media, σ : desviación estándar

Modelo multicanal – Señales X, Z y Altitud				
Clasificación	Exactitud ($\mu \pm \sigma$)%	Sensibilidad ($\mu \pm \sigma$)%	Especificidad ($\mu \pm \sigma$)%	F1 score ($\mu \pm \sigma$)%
Segmentos	61.6±6.6	63.3±8.6	59.9±19.3	62.3±3.4
Sujeto	64.0±9.6	69.1±4.5	58.5±21.3	66.5±5.4

6.5. Comparación de las métricas de desempeño de la validación cruzada en los modelos monocanal y multicanal

Los mejores resultados conseguidos en ambos modelos tienen variaciones que permiten seleccionar el mejor desempeño para un modelo dado. Al analizar los resultados del modelo multicanal agrupando las señales X, Z y Altitud; se obtienen exactitudes de 61.6±6.6% y 64.0±9.6% para la clasificación por segmentos y sujeto, respectivamente. Al comparar este modelo

con el que agrupa todas las señales, se presentan exactitudes superiores al agrupar las tres señales con mejor desempeño en la validación cruzada del monocanal. Por otra parte, el mejor resultado que se consiguió para el modelo monocanal es con la señal de Altitud logrando una exactitud de $61.6 \pm 7.8\%$ para la clasificación por segmentos y $64.0 \pm 8.4\%$ en la clasificación por sujeto, esto se debe a que en principio el ángulo de altitud es el grado de inclinación del lápiz respecto a la superficie de la tablet y esto se ha comprobado en Drotár et al. [41] donde los resultados que obtuvieron demuestran que el movimiento en el aire provocados durante la escritura a mano contiene una cantidad significativa de información relevante para el diagnóstico de la enfermedad de Parkinson. Este resultado también es respaldado en [6].

Finalmente, el mejor resultado entre ambos enfoques se logra en la clasificación por sujeto con el enfoque monocanal debido a que este modelo es capaz de realizar una discriminación entre clases de manera balanceada por sus valores de sensibilidad y especificidad, siendo de $60.9 \pm 20.7\%$ y $66.4 \pm 13.4\%$ respectivamente. En cambio, el enfoque multicanal presenta una sensibilidad del $69.1 \pm 4.5\%$ y especificidad del $58.5 \pm 21.3\%$ lo cual indica un sesgo hacia la detección de pacientes con EP y poca capacidad para clasificar sujetos de CS.

7. CONCLUSIONES

En este trabajo se diseñan dos arquitecturas de redes neuronales convolucionales unidimensionales con dos enfoques distintos, analizando de manera independientes las señales generadas en el proceso de escritura online en un modelo monocanal y combinando las diferentes señales en un modelo multicanal. Ambas con la finalidad de clasificar pacientes con enfermedad de Parkinson. Se utilizaron dos metodologías para encontrar los mejores parámetros de las arquitecturas, una para cada modelo, y posteriormente se evalúa el desempeño de los modelos para la clasificación de pacientes con EP vs CS usando los segmentos generados de los sujetos y aplicando una clasificación por sujeto al calcular el score promedio de los segmentos pertenecientes a cada sujeto. En la clasificación se hace uso de una estrategia de validación cruzada k-fold para verificar la generalización del modelo monocanal y multicanal, con la finalidad de comparar dichos modelos y determinar si la información de las diferentes señales capturadas es complementaria o si por el contrario son redundantes.

Para el modelo monocanal usando la señal de altitud se obtuvo un 61.6% de exactitud tomando la clasificación por segmentos y 64.0% de exactitud tomando la clasificación por sujeto. Para el modelo multicanal combinando las señales X, Z y Altitud se logran los mismo valores de exactitudes en la clasificación por segmentos y clasificación por sujeto, por otra parte, aunque los mejores resultados obtenidos en este trabajo para la clasificación por sujeto son iguales en ambos enfoques, se identificó que la sensibilidad y especificidad en el modelo monocanal presenta menor variación entre sí, lo cual genera una capacidad de clasificación entre clases mejor balanceada en comparación con el mejor modelo multicanal. Los resultados obtenidos usando el enfoque monocanal son sólidos de acuerdo con lo reportado en el estado del arte, donde las señales relacionadas a los movimientos que el sujeto realiza en el aire muestran una gran capacidad para discriminar pacientes con EP de CS. Los resultados del modelo multicanal presentan un desempeño inferior esto puede deberse a que algunas señales no aportan más información, además, aunque las exactitudes de los modelos multicanal exhiben menor variación entre sí en ambos tipos de clasificación haciéndolas más consistentes; la sensibilidad y especificidad obtenidas en el mejor modelo demuestran un sesgo hacia los sujetos con EP.

En general, los resultados obtenidos en este trabajo indican que el análisis del deterioro grafomotor para la detección de la EP basado en 1D-CNN de las diferentes señales dinámicas extraídas en el proceso de escritura online no logran superar los enfoques basados en análisis de imágenes. Esto puede deberse a diversos factores entre ellos la limitada cantidad de datos disponibles para entrenar las arquitecturas donde pueda analizar la información entre los segmentos generados a partir de subsecuencias temporales más cortas, además la inclusión de unidades recurrentes en la arquitectura 1D-CNN como un LSTM (Long Short Term Memory) o un GRU (Gated Recurrent Unit) podría mejorar el desempeño de los modelos. Los trabajos futuros incluirán estrategias de aumento de datos o aprendizaje por transferencia siendo enfoques usados recientemente que han demostrado buena exactitud en la evaluación de la pérdida de habilidades grafomotoras para la detección de la EP.

8. REFERENCIAS

- [1] M. J. Armstrong y M. S. Okun, “Diagnosis and Treatment of Parkinson Disease: A Review”, *JAMA*, vol. 323, núm. 6, p. 548, feb. 2020, doi: 10.1001/jama.2019.22360.
- [2] I. Aouraghe, G. Khaissidi, y M. Mrabti, “A literature review of online handwriting analysis to detect Parkinson’s disease at an early stage”, *Multimed. Tools Appl.*, sep. 2022, doi: 10.1007/s11042-022-13759-2.
- [3] D. J. Surmeier, “Determinants of dopaminergic neuron loss in Parkinson’s disease”, *FEBS J.*, vol. 285, núm. 19, pp. 3657–3668, oct. 2018, doi: 10.1111/febs.14607.
- [4] A. Letanneux, J. Danna, J.-L. Velay, F. Viallet, y S. Pinto, “From micrographia to Parkinson’s disease dysgraphia: Parkinson’s Disease Dysgraphia”, *Mov. Disord.*, vol. 29, núm. 12, pp. 1467–1475, oct. 2014, doi: 10.1002/mds.25990.
- [5] M. Thomas, A. Lenka, y P. Kumar Pal, “Handwriting Analysis in Parkinson’s Disease: Current Status and Future Directions”, *Mov. Disord. Clin. Pract.*, vol. 4, núm. 6, pp. 806–818, nov. 2017, doi: 10.1002/mdc3.12552.
- [6] C. D. Rios-Urrego, J. C. Vásquez-Correa, J. F. Vargas-Bonilla, E. Nöth, F. Lopera, y J. R. Orozco-Aroyave, “Analysis and evaluation of handwriting in patients with Parkinson’s disease using kinematic, geometrical, and non-linear features”, *Comput. Methods Programs Biomed.*, vol. 173, pp. 43–52, may 2019, doi: 10.1016/j.cmpb.2019.03.005.
- [7] U. Gupta, H. Bansal, y D. Joshi, “An improved sex-specific and age-dependent classification model for Parkinson’s diagnosis using handwriting measurement”, *Comput. Methods Programs Biomed.*, vol. 189, p. 105305, jun. 2020, doi: 10.1016/j.cmpb.2019.105305.
- [8] P. Khatamino, I. Canturk, y L. Ozyilmaz, “A Deep Learning-CNN Based System for Medical Diagnosis: An Application on Parkinson’s Disease Handwriting Drawings”, en *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, Istanbul, Turkey, oct. 2018, pp. 1–6. doi: 10.1109/CEIT.2018.8751879.
- [9] S. Chakraborty, S. Aich, Jong-Seong-Sim, E. Han, J. Park, y H.-C. Kim, “Parkinson’s Disease Detection from Spiral and Wave Drawings using Convolutional Neural Networks: A Multistage Classifier Approach”, en *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, Phoenix Park, PyeongChang,, Korea (South), feb. 2020, pp. 298–303. doi: 10.23919/ICACT48636.2020.9061497.

-
- [10] C. R. Pereira *et al.*, “Handwritten dynamics assessment through convolutional neural networks: An application to Parkinson’s disease identification”, *Artif. Intell. Med.*, vol. 87, pp. 67–77, may 2018, doi: 10.1016/j.artmed.2018.04.001.
- [11] I. Kamran, S. Naz, I. Razzak, y M. Imran, “Handwriting dynamics assessment using deep neural network for early identification of Parkinson’s disease”, *Future Gener. Comput. Syst.*, vol. 117, pp. 234–244, abr. 2021, doi: 10.1016/j.future.2020.11.020.
- [12] C. Taleb, M. Khachab, C. Mokbel, y L. Likforman-Sulem, “Visual Representation of Online Handwriting Time Series for Deep Learning Parkinson’s Disease Detection”, en *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Sydney, Australia, sep. 2019, pp. 25–30. doi: 10.1109/ICDARW.2019.50111.
- [13] C. Taleb, L. Likforman-Sulem, C. Mokbel, y M. Khachab, “Detection of Parkinson’s disease from handwriting using deep learning: a comparative study”, *Evol. Intell.*, sep. 2020, doi: 10.1007/s12065-020-00470-0.
- [14] M. W. Berry, A. Mohamed, y Yap Bee Wah, Eds., *Supervised and unsupervised learning for data science*. Cham: Springer, 2020. [En línea]. Disponible en: <https://link.springer.com/book/10.1007/978-3-030-22475-2>
- [15] S. J. Russell, P. Norvig, y E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010. [En línea]. Disponible en: <https://repository.unimal.ac.id/1022/>
- [21] MATLAB, “What Is Deep Learning? | How It Works, Techniques & Applications”, 2022. [En línea]. Disponible en: <https://www.mathworks.com/discovery/deep-learning.html>.
- [17] MATLAB, “What Is a Convolutional Neural Network? | 3 things you need to know”, 2022. <http://bit.ly/3VeVcX5>
- [18] N. Aloysius y M. Geetha, “A review on deep convolutional neural networks”, en *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, abr. 2017, pp. 0588–0592. doi: 10.1109/ICCSP.2017.8286426.
- [19] DeepAI, “Neural Network”, *DeepAI*, el 17 de mayo de 2019. <https://deepai.org/machine-learning-glossary-and-terms/neural-network>
- [20] Pedro Larrañaga, Iñaki Inza, y Abdelmalik Moujahid, “Redes Neuronales”. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco. [En línea]. Disponible en: <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>

-
- [21] S. Kostadinov, “Understanding Backpropagation Algorithm”, *Medium*, el 12 de agosto de 2019. <http://bit.ly/3YFkd0M>
- [22] DataSmarts Español, “¿Qué es un Optimizador y Para Qué Se Usa en Deep Learning?” <https://bit.ly/3WsJLfs>
- [23] Gupta, “A Comprehensive Guide on Deep Learning Optimizers”, el 7 de octubre de 2021. <https://bit.ly/3FBBCi4>
- [24] C. Hansen, “Optimizers Explained - Adam, Momentum and Stochastic Gradient Descent. Machine Learning From Scratch”, diciembre de 2019. <https://mlfromscratch.com/optimizers-explained/>
- [25] I. Goodfellow, Y. Bengio, y A. Courville, *Deep learning*. Cambridge, Massachusetts: The MIT Press, 2016. [En línea]. Disponible en: <http://www.deeplearningbook.org>
- [26] J. Martínez Heras, “Regularización Lasso L1, Ridge L2 y ElasticNet - IArtificial.net”, el 14 de marzo de 2019. <https://www.iartificial.net/regularizacion-lasso-l1-ridge-l2-y-elasticnet/>
- [27] A. Oppermann, “Regularization in Deep Learning — L1, L2, and Dropout”, *Medium*, el 11 de agosto de 2020. <http://bit.ly/3v48meY>
- [28] H. Yadav, “Dropout in Neural Networks”, *Medium*, el 5 de julio de 2022. <https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9>
- [29] S. Albawi, T. A. Mohammed, y S. Al-Zawi, “Understanding of a convolutional neural network”, en *2017 International Conference on Engineering and Technology (ICET)*, Antalya, ago. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
- [30] M. Gurucharan, “Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog”, el 28 de julio de 2022. <https://www.upgrad.com/blog/basic-cnn-architecture/>
- [31] Andrea Perlato, “CNN and Softmax”. <https://www.andreaperlato.com/aipost/cnn-and-softmax/>
- [32] R. Sánchez-Reolid, F. López de la Rosa, M. T. López, y A. Fernández-Caballero, “One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity”, *Biomed. Signal Process. Control*, vol. 71, pp. 103–203, ene. 2022, doi: 10.1016/j.bspc.2021.103203.
- [33] M. Granat, “How to Use Convolutional Neural Networks for Time Series Classification”, *Medium*, el 5 de octubre de 2019. <https://bit.ly/3FHxWqs>

-
- [34] Dive into deep learning, “Padding and Stride”, 2020. https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html
- [35] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, y D. J. Inman, “1D convolutional neural networks and applications: A survey”, *Mech. Syst. Signal Process.*, vol. 151, p. 107398, abr. 2021, doi: 10.1016/j.ymsp.2020.107398.
- [36] R. Vickery, “5 Minute Guide to Cross Validation”, *Medium*, el 27 de septiembre de 2022. <https://towardsdatascience.com/5-minute-guide-to-cross-validation-be3c5b0ae693>
- [37] D. Escobar Grisales, “Demographic information retrieval from text for subject characterization and market segmentation”, Tesis de Maestría, Maestría en Ingeniería de Telecomunicaciones, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022. [En línea]. Disponible en: <https://hdl.handle.net/10495/28986>
- [38] Amazon Machine Learning, “Validación cruzada”. https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/cross-validation.html
- [39] T. Shin, “Understanding the Confusion Matrix and How to Implement it in Python”, *Medium*, el 4 de diciembre de 2021. <http://bit.ly/3GhZsRH>
- [40] S. Narkhede, “Understanding AUC - ROC Curve”, *Medium*, el 15 de junio de 2021. <http://bit.ly/3YHXfpQ>
- [41] P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, y M. Faundez-Zanuy, “Analysis of in-air movement in handwriting: A novel marker for Parkinson’s disease”, *Comput. Methods Programs Biomed.*, vol. 117, núm. 3, pp. 405–411, dic. 2014, doi: 10.1016/j.cmpb.2014.08.007.