

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Análisis de la escritura para el diagnóstico de la enfermedad de Parkinson,
utilizando técnicas de Inteligencia Artificial

TECH-333

Proyecto Integrador

Previo la obtención del Título de:

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

Presentado por:

Diego Eduardo Arteaga Chávez

Carlos Humberto Meneses Murillo

Guayaquil - Ecuador

Año: 2023-2024

Dedicatoria

El presente trabajo es dedicado a mis padres que siempre me han apoyado y a compañeros de carrera que han sido una impulso e inspiración en algunos momentos.

Diego Eduardo Arteaga Chávez

Carlos Humberto Meneses Murillo

Agradecimientos

Agradezco a quiénes han formado parte del gran viaje que representa mi carrera académica. A mis amigos, colegas, maestros y familiares que me han apoyado y guiado por el camino. No olvidaré las enseñanzas que se me han dado y los momentos por los que he pasado junto a mis compañeros en los diferentes proyectos.

Diego Eduardo Arteaga Chávez

Quiero expresar mi sincero agradecimiento a todos mis compañeros de la universidad por hacer que estos años sean inolvidables. A mis amigos de clases, familiares, a quienes conocí a través del club TAWS y a todos aquellos que han dejado una huella imborrable en mi vida. También quiero agradecer a cada uno de mis profesores por sus valiosas enseñanzas y su apoyo en mis proyectos.

Carlos Humberto Meneses Murillo

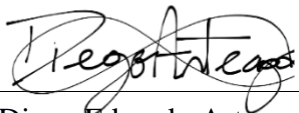
Declaración Expresa

Nosotros, Diego Eduardo Arteaga Chávez y Carlos Humberto Meneses Murillo acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 15 de febrero del 2024.



Diego Eduardo Arteaga

Chávez



Carlos Humberto

Meneses Murillo

Evaluadores

Ph.D. Lucia Marisol Villacres Falconi

Profesor de Materia



firmado electrónicamente por:
**COLÓN ENRIQUE
PELAEZ JARRIN**

Enrique Peláez Jarrín, Ph.D.

Tutor de proyecto

RESUMEN

En la actualidad, la enfermedad de Parkinson ocupa el segundo lugar como la enfermedad neurodegenerativa más común, después del Alzheimer, afectando a aproximadamente 10 millones de personas en todo el mundo. A pesar de no existir una cura definitiva, existen tratamientos disponibles para aliviar los síntomas y mejorar la calidad de vida de los pacientes. Por esto, es esencial una detección temprano de la enfermedad para ralentizar su progresión. Uno de los síntomas característicos de esta afección es la dificultad en la escritura, lo que a menudo motiva a los pacientes a buscar atención médica. Basándose en esta premisa, se propone la utilización de técnicas de aprendizaje poco profundo y aprendizaje profundo para identificar estos cambios en la escritura como patrones asociados a la enfermedad y calcular la probabilidad de que una persona tenga Parkinson.

Para el desarrollo de este proyecto, se utilizó un conjunto de datos que incluía registros de tres grupos diferentes: personas sanas, personas con Parkinson en el hemisferio derecho y personas con Parkinson en el hemisferio izquierdo. Estos registros comprendían la escritura en cuatro experimentos distintos: la firma, la escritura de una palabra en español, polca y griego. A partir de estas escrituras, se extrajeron características matemáticas que se utilizaron como datos de entrenamiento para un modelo SVM, el cual obtuvo los mejores resultados en comparación con otras arquitecturas evaluadas, logrando una tasa de clasificación binaria del 88.06% y un 74% para la clasificación multiclase. Este modelo se implementó en una aplicación web que permite a los usuarios cargar archivos de escrituras de pacientes y obtener un diagnóstico de la enfermedad de Parkinson.

Palabras Clave: Enfermedad de Parkinson, inteligencia artificial, aprendizaje poco profundo, micrografía.

ABSTRACT

Currently, Parkinson's disease ranks as the second most common neurodegenerative disease, following Alzheimer's, affecting approximately 10 million people worldwide. Despite the absence of a definitive cure, there are available treatments to alleviate symptoms and enhance patients' quality of life. Therefore, early detection of the disease is crucial to slow its progression. One of its characteristic symptoms is handwriting difficulty, often prompting patients to seek medical attention. Building on this premise, the use of shallow and deep learning techniques is proposed to identify these handwriting changes as disease-associated patterns and calculate the likelihood of a person having Parkinson's.

To develop this project, a dataset was used, including records from three different groups: healthy individuals, individuals with Parkinson's in the right hemisphere, and individuals with Parkinson's in the left hemisphere. These records encompassed handwriting in four distinct experiments: signature, writing a word in Spanish, polka, and Greek. Mathematical features were extracted from these writings and employed as training data for an SVM model, which outperformed other evaluated architectures, achieving an 88.06% binary classification rate and 74% for multiclass classification. This model was implemented in a web application that enables users to upload patient handwriting files and obtain a Parkinson's disease diagnosis.

Keywords: Parkinson's Disease, artificial intelligence, shallow learning, micrographia.

Índice general

Evaluadores.....	5
Resumen	6
Abstract.....	7
Índice general.....	8
Índice de figuras	11
Índice de tablas	13
CAPÍTULO 1.....	1
1.1 Introducción	1
1.2 Descripción del Problema	2
1.3 Justificación del Problema	4
1.4 Objetivos	5
1.4.1 Objetivo General.....	5
1.4.2 Objetivos específicos	5
1.5 Marco teórico	5
1.5.1 Enfermedad de Parkinson	5
1.5.2 Correlación entre la enfermedad del Parkinson y la escritura a mano	7
1.5.3 Procedimientos actuales para el diagnóstico.....	10
1.5.4 Conceptos importantes en Aprendizaje Automático.....	11
1.5.5 Tecnologías Implementadas.....	12

CAPÍTULO 2.....	15
2. Metodología.....	15
2.1 Obtención de datos de escritura	15
2.2 Preprocesamiento de datos	17
2.3 Desarrollo de modelos a utilizar.....	19
2.3.1 Pruebas con Modelos de Aprendizaje Poco Profundo	19
2.3.2 Pruebas en Modelos de Aprendizaje Profundo	24
2.4 Desarrollo de Prototipo	39
2.4.1 Actores del Sistema.....	39
2.4.2 Requerimientos no funcionales.....	40
2.4.3 Alcance de la solución	41
2.4.4 Riesgos y beneficios del prototipo de la solución.....	41
2.4.5 Arquitectura del prototipo.....	42
2.4.6 Prototipo Actual	44
CAPÍTULO 3.....	47
3. Resultados.....	47
3.1 Experimentos Realizados	47
3.1.1 Entorno de Desarrollo	47
3.2 Comparación de modelos	48
3.2.1 Modelos de Aprendizaje Poco Profundo	48

3.2.2	Modelos de Aprendizaje Profundo	54
3.2.3	Comparación entre modelos y selección del mejor modelo	56
3.3	Pruebas realizadas a la página web	62
CAPÍTULO 4.....		66
4.	Conclusiones y recomendaciones	66
4.1	Conclusiones	67
4.1	Recomendaciones.....	68
ANEXOS		69
BIBLIOGRAFÍA		71

Índice de figuras

Ilustración 1	3
Ilustración 2	14
Ilustración 3	15
Ilustración 4	16
Ilustración 5	16
Ilustración 6	16
Ilustración 7	16
Ilustración 8	21
Ilustración 9	22
Ilustración 10	23
Ilustración 11	24
Ilustración 12	28
Ilustración 13	29
Ilustración 14	31
Ilustración 15	32
Ilustración 16	34
Ilustración 17	36
Ilustración 18	38
Ilustración 19	43
Ilustración 20	44
Ilustración 21	45
Ilustración 22	45

Ilustración 23	46
Ilustración 24	46
Ilustración 25	56
Ilustración 26	59
Ilustración 27	61
Ilustración 28	61
Ilustración 29	63
Ilustración 30	65

Índice de tablas

Tabla 1	29
Tabla 2	30
Tabla 3	31
Tabla 4	33
Tabla 5	33
Tabla 6	35
Tabla 7	35
Tabla 8	37
Tabla 9	37
Tabla 10	39
Tabla 11	39
Tabla 12	48
Tabla 13	50
Tabla 14	51
Tabla 15	52
Tabla 16	53
Tabla 17	54
Tabla 18	57
Tabla 19	58
Tabla 20	60
Tabla 21	62
Tabla 22	64

CAPÍTULO 1

1.1 Introducción

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo crónico que impacta considerablemente en la calidad de vida de las personas que la padecen. Este trastorno se caracteriza por una serie de síntomas motores, que incluyen temblores en reposo, bradicinesia, rigidez e inestabilidad postural [50]. Según datos de la Fundación de Parkinson, se estima que aproximadamente 10 millones de personas en todo el mundo conviven con esta enfermedad [40]. Aunque la edad media de inicio de los síntomas suele situarse en torno a los 60 años, es importante destacar que un 10% de los diagnosticados son menores de 50 años [40].

Los síntomas motores comienzan a aparecer una vez que los cuerpos de Lewy (depósitos anormales de proteína llamada alfa-sinucleína) se han propagado a la sustancia negra según la hipótesis de Braak [47]. Los criterios actuales para diagnosticar la enfermedad de Parkinson incluyen la necesidad de observar bradicinesia junto con rigidez o temblor [47]. Aunque la inestabilidad postural es una característica central de esta enfermedad, si se presentan caídas tempranas, se debe considerar la posibilidad de un síndrome parkinsoniano poco común, como la parálisis supranuclear progresiva (PSP) [47]. La detección temprana de la enfermedad es crucial, ya que los esfuerzos para desarrollar terapias neuro protectoras se enfocan en las etapas iniciales de la EP, las cuales ofrecen la mejor oportunidad para intervenir [48].

En tiempos recientes, se han desarrollado soluciones innovadoras que emplean inteligencia artificial para la detección temprana y el monitoreo de los síntomas de la Enfermedad de Parkinson (EP). Por ejemplo, en un estudio reciente se utilizaron tres algoritmos de clasificación: SVM, K-NN y AdaBoost, con el propósito de construir un modelo predictivo de la EP a partir de

características cinemáticas y de presión relacionadas con la escritura [42]. Se aplicaron pruebas estadísticas para seleccionar las 20 características más relevantes. Los resultados mostraron que el modelo SVM alcanzó la mayor precisión de clasificación, alcanzando un impresionante 82% de precisión utilizando únicamente características cinemáticas y de presión básicas [42]. El modelo K-NN logró una precisión de clasificación del 71.7% con un valor de k igual a 3, mientras que el modelo AdaBoost obtuvo una precisión de clasificación del 78.9% utilizando árboles de decisión como clasificadores débiles y deteniéndose después de 500 estimadores.

El presente trabajo se centra en el análisis de la escritura en pacientes con EP utilizando un modelo predictivo que es escogido tras la comparación entre el mejor modelo que hace uso exclusivo de datos tabulares basados en las características extraídas de las muestras de escritura y el mejor modelo que hace uso exclusivo de imágenes de muestras de escritura. El enfoque del proyecto es de naturaleza no intrusiva, lo cual puede resultar útil para identificar el deterioro en la escritura de las personas, ya que frecuentemente constituye uno de los primeros signos que motiva la búsqueda de atención médica [45]. De hecho, se ha observado que aproximadamente el 5% de los pacientes con EP experimentan dificultades en la escritura antes de manifestar cualquier otro síntoma motor, y después del inicio de los síntomas, el 30% informa acerca del agravamiento de sus habilidades de escritura [45].

1.2 Descripción del Problema

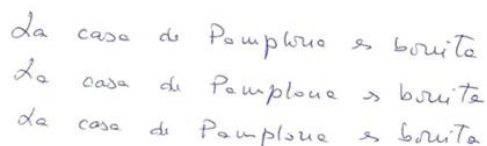
La identificación automática de la EP en fases tempranas es un desafío para la medicina actual, ya que el comportamiento en las primeras etapas suele ser similar a personas sanas [58]. Aunque la escritura no se incluye formalmente en los diagnósticos de la EP, el empeoramiento de esta es una de las principales razones por lo que los pacientes buscan atención médica [59]. No obstante, el diagnóstico y el estudio de estos síntomas motores se complican y se vuelven menos

detectables en pacientes donde el lado dominante no coincide con el inicio de los síntomas motores en la EP [60].

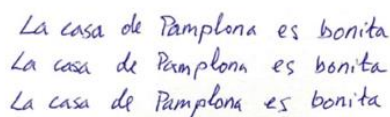
La escritura se revela como una característica que puede identificarse desde las primeras etapas de esta enfermedad, lo que subraya la importancia de su análisis para detectar signos tempranos y permitir un tratamiento oportuno [54]. No obstante, estas alteraciones pueden llegar a ser casi imperceptibles en un inicio para las personas y doctores, por lo que se podría usar las bondades de la IA para analizar la escritura de una persona que se encuentra afectada por diferentes hemisferios en esta enfermedad [61]. Gracias a la calidad de los dispositivos de comunicación disponibles actualmente, fáciles de usar, no es necesario que el diagnóstico se haga solo en un entorno hospitalario. Esto abre la puerta a un diagnóstico más accesible y conveniente sin necesidad de movilizarse a un centro médico [61].

Ilustración 1

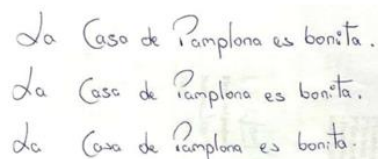
Muestras de escritura en papel, controles vs pacientes



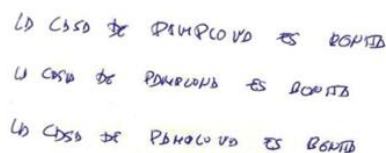
La casa de Pamplona es bonita
La casa de Pamplona es bonita
La casa de Pamplona es bonita



La casa de Pamplona es bonita
La casa de Pamplona es bonita
La casa de Pamplona es bonita



La Casa de Pamplona es bonita.
La Casa de Pamplona es bonita.
La Casa de Pamplona es bonita.



La Casa de Pamplona es bonita.
La Casa de Pamplona es bonita.
La Casa de Pamplona es bonita.

Nota. Extraído de [65].

1.3 Justificación del Problema

La epidemiología de la enfermedad de Parkinson presenta incertidumbres al realizar un diagnóstico preciso en vida [62]. Con el envejecimiento de la población y el aumento de expectativa que existe de vida, se espera que haya un incremento significativo en el número de casos para las próximas décadas [62]. La Organización Mundial de la Salud proyectó que para el año 2040 existirá alrededor de 17 millones de casos en todo el mundo [46]. Por ello, el proyecto presente se enfoca en el análisis de datos de escritura a mano para desarrollar un modelo de inteligencia artificial que sea capaz de diagnosticar de forma prematura la enfermedad de Parkinson, porque las actuales técnicas utilizadas para evaluar la enfermedad dependen en gran medida de la pericia de los expertos [43]. Es menester señalar que, dado que la enfermedad afecta de forma crónica y progresiva al paciente que la padece, la negligencia que puede existir al aplicar las técnicas mencionadas en el diagnóstico y monitoreo podría traer un impacto negativo en los pacientes en términos de costos de asistencia médica y terribles desordenes relacionados a la salud [44]. Los enfoques no intrusivos son más apropiados para ancianos y no requieren la constante intervención de especialistas. Entre las medidas no invasivas tenemos el trazado de formas como espirales, ondas y otros textos escritos a mano [44]. Tradicionalmente para las medidas mencionadas, se necesitaba un intérprete, lo cual podía representar una desventaja, pero en el presente proyecto que tiene el enfoque no intrusivo junto a sus beneficios, se hará uso de una tableta para obtener los datos de escritura de forma más eficiente [51].

En términos prácticos, este proyecto se centra en el análisis de la escritura como una herramienta fundamental para poder realizar el diagnóstico temprano de la enfermedad de Parkinson debido a una serie de ventajas sustanciales. La fácil accesibilidad y lo poco invasivo

que es la escritura permite que este sea un enfoque idóneo, ya que puede ser implementado de manera sencilla y sin tener que requerir costosos procedimientos médicos [44].

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un prototipo que utilice técnicas de Aprendizaje Profundo o Aprendizaje Poco Profundo para asistir a los profesionales de la salud en la identificación temprana de la enfermedad de Parkinson a través del análisis de la escritura del paciente. Este sistema analizará muestras de escritura de pacientes previamente diagnosticados con la enfermedad, así como de individuos sanos que servirán como grupo de control.

1.4.2 Objetivos específicos

- Identificar las características importantes en el deterioro de la escritura.
- Clasificar patrones asociados al deterioro de la escritura de estos pacientes
- Entrenar y evaluar los modelos de Aprendizaje Poco Profundo y Aprendizaje Profundo utilizando la base de datos recopilada, utilizando métricas de rendimiento adecuadas.
- Identificar el mejor modelo de IA para predecir la probabilidad de que una persona padezca la EP.

1.5 Marco teórico

1.5.1 Enfermedad de Parkinson

La enfermedad de Parkinson (EP) es una compleja enfermedad neurodegenerativa que afecta a una gran parte de la población mundial [1]. Con tasas de prevalencia actuales que varían entre 10 y 800 personas por cada 100,000 habitantes, la EP está entre los trastornos

neurodegenerativos más comunes [2]. Se caracteriza por ser un trastorno del movimiento que se manifiesta por medio de síntomas como temblor en reposo, rigidez, pérdida de reflejos posturales

El diagnóstico de la Enfermedad de Parkinson (EP) se basa en la evaluación de las principales manifestaciones clínicas que se manifiestan en los pacientes, que incluyen [5]:

- Manifestaciones cardinales: Estas son las características fundamentales de la enfermedad e incluyen el temblor en reposo, la rigidez muscular, la lentitud de movimiento (acinesia) y la inestabilidad postural.
- Manifestaciones no motoras: Además de los síntomas motores, la EP también se asocia con manifestaciones no motoras que pueden incluir déficits cognitivos sin llegar a la demencia, trastornos del estado de ánimo como la depresión y la ansiedad, así como problemas relacionados con el sueño.
- Alteraciones en los ojos: La enfermedad puede afectar la función ocular, lo que puede dar lugar a problemas visuales y oculares.
- Disartria: La disartria es un trastorno del habla que se caracteriza por dificultades en la articulación de palabras debido a la afectación de los músculos responsables del habla.
- Disfagia: La disfagia se refiere a dificultades en la deglución, lo que puede ocasionar problemas al tragar alimentos y líquidos.
- Anomalías olfatorias: Los pacientes con EP pueden experimentar alteraciones en el sentido del olfato.
- Alteraciones autonómicas: Estas manifestaciones pueden incluir problemas como el estreñimiento, la hipotensión ortostática (descenso de la presión arterial al ponerse de pie), disfunción sexual y trastornos urinarios.

- Sensitivas: Los pacientes pueden experimentar dolor, parestesias (sensaciones anormales como hormigueo) y calambres.
- Dermatológicas: Algunas manifestaciones cutáneas, como la seborrea (exceso de producción de grasa en la piel), también pueden estar presentes en la EP.

Uno de los mayores obstáculos en el diagnóstico de la Enfermedad de Parkinson (EP) radica en la carencia de un método objetivo para su detección. Actualmente, el diagnóstico puede prolongarse varios meses y requiere una vigilancia meticulosa de los síntomas. Incluso con este seguimiento riguroso, existe una probabilidad aproximada del 25% de recibir un diagnóstico erróneo [52].

1.5.2 Correlación entre la enfermedad del Parkinson y la escritura a mano

Trastornos neurológicos como el autismo [6,7], el Parkinson [9] y el Alzheimer [11,20] tienen un impacto directo en las habilidades grafomotoras de las personas afectadas. Además de estos trastornos, se ha observado que la influencia en la escritura también puede ser causada por medicamentos psicotrópicos [21] y el proceso de envejecimiento [22,23]. Los efectos visibles de trastornos neurológicos, como la apraxia constructiva, la disgrafía y la micrografía, se pueden evaluar directamente mediante el análisis de muestras de escritura y dibujos a mano de los individuos [53].

Las tres principales manifestaciones de la EP que pueden ser detectadas a través de la escritura a mano incluyen la micrografía, la bradicinesia y el temblor [17]. La micrografía se refiere a la dificultad que experimentan los pacientes para mantener un tamaño y alineación adecuados en las letras y trazos que producen al escribir [25]. La bradicinesia, que se caracteriza por la lentitud en los movimientos, hace que los pacientes con EP requieran más tiempo del habitual para

completar tareas de escritura y dibujo [26]. Por último, los "temblores" se refieren a movimientos involuntarios de vaivén que se manifiestan en la escritura a través de trazos irregulares en caracteres y dibujos. Estas condiciones parkinsonianas pueden aparecer de manera independiente o coexistir, dependiendo del tipo y la etapa de la enfermedad [54]. Identificar con precisión estos indicadores en muestras de escritura de pacientes puede ser de gran ayuda en la detección temprana y en el diagnóstico diferencial de la EP [54].

Investigaciones recientes han planteado la posibilidad de utilizar la escritura a mano como un indicador para el diagnóstico de la Enfermedad de Parkinson (EP) [12,13]. La razón subyacente a esta idea es que la escritura normal se caracteriza por la automatización; los movimientos son tan rápidos que el bucle de retroalimentación normal a través de la percepción visual y el control muscular se desactiva. Esto resulta en una configuración de bucle abierto. En el caso de los pacientes con Parkinson, la automatización ya no es válida; su escritura depende del bucle cerrado visual [14]. Esto sugiere que las personas con EP confían más en la retroalimentación visual para guiar su escritura, posiblemente debido a alteraciones en su control motor. En contraste, las personas sanas pueden escribir de manera más automática y requieren menos retroalimentación visual.

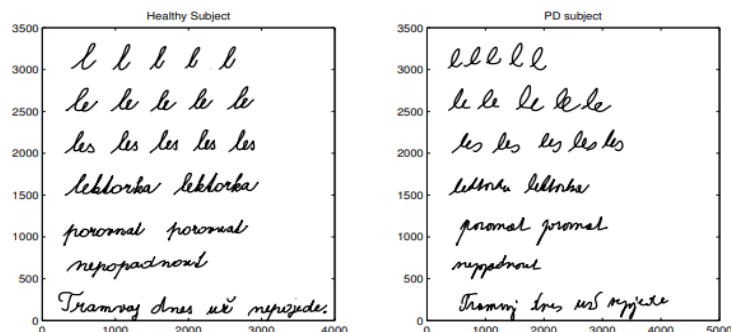
En la literatura científica [55-57], se han propuesto diversas características de la escritura que pueden utilizarse para predecir la presencia de la Enfermedad de Parkinson (EP) y otros trastornos neurológicos. Estas características se pueden dividir en dos categorías según cómo se obtiene y mide la información: características estáticas y características dinámicas. Las características estáticas se obtienen típicamente a partir de muestras de escritura que no requieren de una conexión en tiempo real y se centran en atributos espaciales, como la forma y las dimensiones de los caracteres (por ejemplo, sesgo e inclinación), proporciones (como

altura/anchura y relación de aspecto), distancias entre caracteres, colocación en el plano x-y, bucles, cierres y curvatura de los trazos. Estas características se han utilizado en investigaciones previas para identificar signos de micrografía y temblores en la escritura [27]. Por otro lado, las características dinámicas requieren información temporal adicional y se obtienen a partir de muestras de escritura en línea, que capturan los movimientos de escritura y medidas cinemáticas como la velocidad y la aceleración del trazo en dirección vertical y horizontal, así como las trayectorias del movimiento. Estas características dinámicas se obtienen comúnmente utilizando dispositivos como tabletas digitalizadoras y bolígrafos inteligentes [28]. Además, estos dispositivos pueden registrar intervalos de tiempo en el aire o en la superficie, que reflejan el tiempo que un individuo tarda en planificar la acción de escritura siguiente, siendo un indicador de la capacidad cognitiva [28]. La presión ejercida en el instrumento de escritura también es un indicador relevante, ya que disminuye con la progresión de la EP [29]. Para medir con mayor precisión la presión del bolígrafo, se utilizan dispositivos especializados en la adquisición de muestras de escritura en línea.

En el contexto de las pruebas de evaluación de la EP, la representación de espirales se ha utilizado ampliamente, ya que permite evaluar el rendimiento motor en diversos trastornos del movimiento, incluyendo la EP [4,8,15-17]. Además, se han empleado palabras que contienen repeticiones de la letra cursiva "l" en la evaluación de la escritura [10,18].

Ilustración 2

Presencia de la letra cursiva "l" en la evaluación de escritura



Nota. Extraído de [55]

1.5.3 Procedimientos actuales para el diagnóstico

Actualmente el diagnóstico de la EP se basa en una combinación de la historia médica, el examen físico y, en algunos casos, pruebas de apoyo como imágenes cerebrales y pruebas de función dopaminérgica. La confirmación del diagnóstico se realiza en consulta con un médico especializado en trastornos del movimiento [49].

La Sociedad Internacional de Parkinson y Trastornos del Movimiento (MDS) ha desarrollado los criterios recientes para el diagnóstico de la EP, que reflejan el conocimiento actual de la enfermedad. Para que se considere un diagnóstico de EP, el paciente debe presentar bradicinesia (lentitud de movimiento) y al menos uno de los siguientes síntomas: temblor en un miembro en reposo, rigidez en brazos, piernas o tronco, problemas de equilibrio y caídas [49].

Una de las pruebas que pueden contribuir a respaldar el diagnóstico de la EP es el DaTscan [49], que evalúa la función del sistema de dopamina en el cerebro. Aunque el DaTscan puede resultar útil para confirmar la presencia de la EP, no se considera necesario en todos los casos. Un DaTscan positivo puede ayudar a confirmar la enfermedad de Parkinson. Sin embargo, un DaTscan negativo no descarta la enfermedad de Parkinson, especialmente en las primeras etapas de la

enfermedad [49]. En última instancia, el diagnóstico se basa principalmente en la evaluación clínica y los síntomas del paciente, y el DaTscan se emplea como una herramienta complementaria en situaciones específicas.

1.5.4 Conceptos importantes en Aprendizaje Automático

En el ámbito del aprendizaje de representación de datos, los algoritmos se pueden clasificar en dos categorías: aprendizaje poco profundo y aprendizaje profundo, según el nivel de abstracción jerárquica de los datos [89].

El aprendizaje de modelos poco profundos es un tipo de algoritmos de aprendizaje automático que puede generar un buen modelo predictivo generalizado con solo unas pocas capas de composición. Requiere muestras con características discriminativas bien estudiadas extraídas por expertos [90]. Ejemplos de algoritmos de aprendizaje poco profundo incluyen la Regresión Lineal, la Regresión Logística, los Árboles de Decisión, KNN (K-Nearest Neighbors) y SVM (Support Vector Machines) [88].

El aprendizaje profundo en sí mismo es una rama del aprendizaje automático, que puede entenderse como redes neuronales con múltiples capas ocultas. En comparación con las aplicaciones basadas en aprendizaje poco profundo, los modelos de aprendizaje profundo requieren grandes cantidades de datos de entrenamiento [88].

En el ámbito de la construcción, el aprendizaje profundo presenta ventajas notables al procesar datos complejos y no estructurados, como imágenes y videos, permitiendo la identificación precisa de patrones en grandes conjuntos de datos. Sin embargo, su desventaja radica en la necesidad de grandes cantidades de datos de entrenamiento y recursos computacionales significativos. Por otro lado, el aprendizaje poco profundo, representado por técnicas como la

Regresión logística, Árboles de Decisión y SVM, destaca por su simplicidad y capacidad de interpretación, siendo más eficiente en situaciones con conjuntos de datos más pequeños [88]. Aunque estas técnicas son menos propensas a problemas como el sobreajuste, también pueden presentar limitaciones en términos de precisión y capacidad para manejar datos faltantes [88]. La elección entre aprendizaje profundo y poco profundo dependerá de las demandas específicas del problema y de la disponibilidad de recursos y datos de entrenamiento en el contexto de la construcción. [88]

1.5.5 Tecnologías Implementadas

En una serie de estudios relacionados [30,31,39], se examinó la eficacia de diversas características en línea para predecir la EP. La mayoría de estas características se centraban en mediciones cinemáticas, como la velocidad del trazo, la aceleración en direcciones vertical y horizontal, y la sacudida, que se calculaban a partir de intervalos de tiempo en el aire y en la superficie [55].

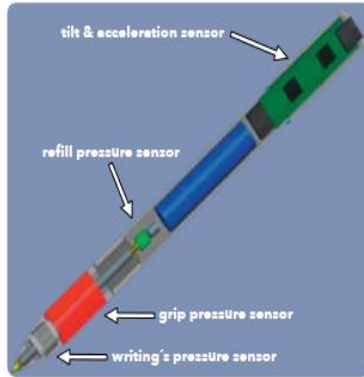
En dos estudios [34,35], se evaluaron características tanto offline como en línea basadas en señales (datos registrados o medidas relacionadas con la escritura o el movimiento de una persona) para predecir la EP utilizando diversos clasificadores de grafos y aprendizaje automático. Se utilizó datos con muestras de espirales dibujadas a mano por 55 individuos para evaluar las características offline, y se alcanzó una precisión de aproximadamente el 75% combinando diferentes características offline con un clasificador SVM. También se evaluaron características basadas en señales utilizando un conjunto de datos de dibujos en espiral y el curso de trazado de 35 sujetos, y las precisiones oscilaron entre el 79% y el 87%. Sin embargo, se observó que los resultados variaban según el clasificador utilizado.

En el artículo de Pereira et al. [63], se alcanzó una precisión del 83.77% con el método OPF (Optimum-Path Forest) utilizando imágenes de 64×64 píxeles y la mitad del conjunto de datos para el entrenamiento. También se utilizaron Redes Neuronales Convolucionales (CNN), la arquitectura ImageNet logró una precisión general del 87.14% con una división de entrenamiento/prueba del 75% / 25% utilizando imágenes de 128×128 píxeles, mientras que la arquitectura LeNet tuvo las tasas de reconocimiento más bajas [63]. Estos experimentos se llevaron a cabo utilizando la biblioteca Caffe [119] e involucraron diversas arquitecturas de CNN, con un análisis estadístico realizado mediante validación cruzada en 20 ejecuciones [63]. Se emplearon características basadas en la dinámica de la escritura a mano, específicamente dibujos de escritura realizados por personas sanas y pacientes, con el objetivo de contribuir al diagnóstico de la enfermedad de Parkinson [63].

En general, se observa una tendencia creciente hacia el uso de características dinámicas en línea debido a su facilidad de adquisición y la riqueza de información que proporcionan. A pesar de esto, no se debe subestimar el potencial de la información visual estática. Es fundamental considerar diversos parámetros, como la tarea de escritura seleccionada y las características técnicas de clasificación, para evaluar el rendimiento de un sistema de predicción automatizado.

Ilustración 2

Pluma Biométrica - sensores localizados en cuatro puntos diferentes



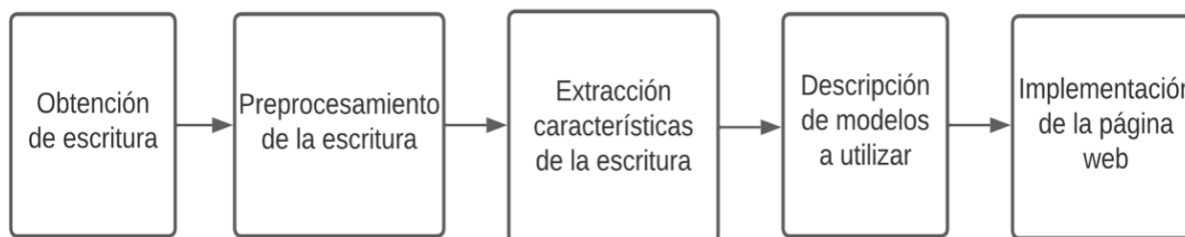
Nota. Extraído de [63].

CAPÍTULO 2

En este capítulo se detalla la metodología, la cual está dividida en 5 etapas secuenciales y que se pueden observar en la **Ilustración 3**, son: obtención de escritura, preprocesamiento de la escritura, extracción de características de la escritura, diseño e implementación de modelos basados en aprendizaje autónomo a utilizar en el sistema. Cada una de estas etapas es importante para la realización del proyecto y es descrita en su respectivo procedimiento.

Ilustración 3

Etapas de la metodología aplicada




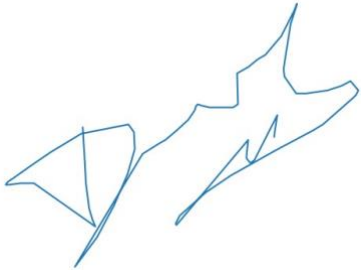


2. Metodología

2.1 Obtención de datos de escritura

El conjunto de datos se obtuvo a través de experimentos realizados en individuos sanos, así como en aquellos que padecían EP predominante en el hemisferio derecho e izquierdo. La tableta SMART TAB se empleó para estos experimentos, proporcionando una interfaz versátil para la tarea de escritura. Su avanzada capacidad de entrada/salida (I/O) y control de adquisición de datos permitieron una marcación continua y sincronización con otros dispositivos. La SMART TAB ofreció control total sobre la configuración de adquisición de datos y una exportación eficiente para análisis detallado. Durante los experimentos, se llevaron a cabo cuatro tipos de

pruebas diferentes: la firma (**Ilustración 7**), la escritura de una palabra en español (**Ilustración 4**), polaco (**Ilustración 6**) y griego (**Ilustración 5**). En resumen, el conjunto de datos se dividió en 1,160 registros de personas sanas, 920 registros de personas con Parkinson en el hemisferio derecho y 600 registros de personas con Parkinson en el hemisferio izquierdo.

<p>Ilustración 4</p> <p><i>Palabra en español</i></p> 	<p>Ilustración 5</p> <p><i>Palabra en griego</i></p> 
<p>Ilustración 6</p> <p><i>Palabra en polaco</i></p> 	<p>Ilustración 7</p> <p><i>Firma</i></p> 

2.2 Preprocesamiento de datos

Inicialmente, se adquirieron datos a partir de archivos de escritura, de donde se extrajo información clave como, series temporales, coordenadas en los ejes x, y, el tipo de evento de prueba, y la afiliación al grupo de sujetos. Las coordenadas extraídas corresponden a los puntos por lo que pasó el lápiz durante cada experimento. Estos datos se almacenaron en archivos en formato MATLAB. En algunos de estos archivos, se identificó discrepancias entre el número de registros de tiempo y las coordenadas registradas, lo que requirió la eliminación de las últimas entradas de tiempo para lograr una concordancia.

Para los modelos de aprendizaje poco profundo, se crearon tres conjuntos de datos distintos: el primero con características médicas, el segundo con características matemáticas, y un tercer conjunto que combinaba las características de los conjuntos previos. Posterior a eso, se extrajeron las características más importantes.

Para el conjunto de características matemáticas, se empleó la biblioteca “tsfresh” [107], la cual genera una amplia variedad de características de series temporales. En el proceso de preparación de datos para el análisis y modelado con “tsfresh”, se sigue una serie de pasos. Primero, se crea un DataFrame con columnas para 'id', 'time', 'x', y 'y', que representan las series temporales de coordenadas. Luego, se realiza la conversión de esas columnas a tipos de datos enteros para asegurar la compatibilidad con las operaciones de “tsfresh”. A continuación, se utiliza la función “extract_features” de “tsfresh” para derivar un nuevo DataFrame que contiene diversas características temporales generadas a partir de las series temporales proporcionadas. Se crea una serie (instancia de la clase Pandas Series) para almacenar las etiquetas relacionadas con la presencia de Parkinson. La imputación de posibles valores nulos en el DataFrame de características extraídas se realiza mediante la función `impute` de “tsfresh”. Finalmente, se utiliza la función

“select_features” de “tsfresh” para realizar la selección de características, considerando su contribución a la variabilidad en las etiquetas de presencia de Parkinson. El resultado es un nuevo DataFrame con 761 características matemáticas.

Las características médicas abarcaban diversos parámetros relacionados con la posición horizontal y vertical, así como con la velocidad, aceleración y jerk. Además, se incluyeron métricas de trazo, como el NCV (número de cambios de velocidad por trazado) y el NCA (número de cambios de aceleración por trazado) [64]. Para enriquecer estas características médicas, se aplicaron siete medidas estadísticas, que incluyen la media, mediana, desviación estándar, valor máximo, rango intercuartil, kurtosis y sesgo (skewness). Inicialmente, se calcularon métricas básicas a lo largo de las dimensiones 'x', 'y', almacenándolas en listas que luego se utilizaron para construir un DataFrame. Posteriormente, se realizaron operaciones para convertir las columnas velocidad, aceleración y jerk a valores absolutos, seguidas por el cálculo de estadísticas adicionales para cada dimensión. El resultado es un DataFrame que incorpora métricas cinemáticas derivadas de las coordenadas de las series temporales, enriquecidas con características estadísticas que proporcionan una representación más completa de los datos médicos.

Con los tres conjuntos de datos, se aplicaron métodos de extracción de características para cada uno de estos. Los métodos utilizados fueron Recursive Feature Elimination (RFE) [108], la prueba U de Mann-Whitney [109] y la prueba de Kruskal-Wallis [110]. Con estos métodos se extrajeron las 10, 15, 20 y 25 características más importantes y, para cada uno de estos subconjuntos de características, se generó un archivo en formato CSV.

Para los modelos de aprendizaje profundo, se implementó una función que, para cada muestra de escritura, emplea sus correspondientes coordenadas que se filtran para crear una representación gráfica de las coordenadas. Tras la creación del gráfico, se invierte su eje para lograr

obtener una correcta representación de la muestra de escritura. Cada gráfico se almacenó como un archivo en formato PNG. Posteriormente, se introdujo una funcionalidad adicional en la cual, para cada imagen, se utilizó la biblioteca OpenCV para leer, redimensionar y comprimir la imagen en formato JPEG. Además, se realizó el etiquetado de las imágenes según su grupo y según la condición del sujeto de prueba con respecto a la EP, para ello utilizó la codificación one-hot de las etiquetas. Finalmente, crea registros TFRecord utilizando TensorFlow, que contienen las imágenes procesadas y sus respectivas etiquetas.

2.3 Desarrollo de modelos a utilizar

En la etapa del desarrollo del proyecto, se realizaron pruebas exhaustivas en modelos de predicción basados en aprendizaje poco profundo con los datos tabulares producidos en la etapa anterior y en modelos de aprendizaje profundo con las imágenes generadas de la escritura. Posteriormente, se llevó a cabo un análisis comparativo para determinar el modelo más efectivo.

2.3.1 Pruebas con Modelos de Aprendizaje Poco Profundo

En esa etapa de la investigación, se llevó a cabo pruebas en modelos de aprendizaje poco profundo para evaluar el rendimiento de los datos de atributos extraídos. Cada modelo fue entrenado utilizando todos los atributos disponibles para cada uno de los conjuntos de datos, así como los subconjuntos de características importantes de cada uno de estos.

Durante el entrenamiento y las pruebas, se monitoreó la precisión y la pérdida, registrando estos datos a lo largo de las épocas. Los resultados fueron graficados mediante diagramas de líneas para analizar el comportamiento de estas métricas a lo largo del tiempo.

Se realizaron pruebas específicas para predecir la condición del paciente y para asignarlos a grupos basados en la afectación en la zona izquierda o derecha del cuerpo, o en el grupo de

pacientes sanos. Estas pruebas abordaron la complejidad de pacientes diestros cuya zona izquierda se vio más afectada por la EP.

La evaluación del rendimiento de los modelos incluyó el análisis de matrices de confusión, proporcionando una visión detallada de los resultados de predicción y permitiendo evaluar la precisión, sensibilidad y especificidad de los modelos en diferentes escenarios.

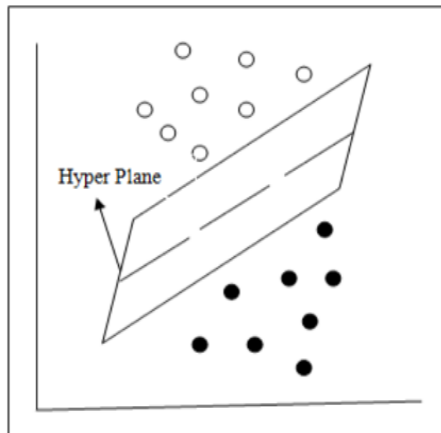
Los modelos poco profundos seleccionados para la evaluación del rendimiento de los datos extraídos incluyen Support Vector Machine (SVM), AdaBoost, Random Forests y Multi-Layer Perceptron (MLP).

2.3.1.1 Support Vector Machine (SVM)

SVM es un algoritmo de aprendizaje supervisado usado tanto para clasificación como para regresión [66]. La idea principal detrás de las Máquinas de Vectores de Soporte (SVM) es la construcción de un hiperplano óptimo que se puede utilizar para la clasificación de patrones linealmente separables. El hiperplano óptimo es seleccionado de un conjunto de hiperplanos para clasificar patrones, maximizando el margen del hiperplano, es decir, la distancia desde el hiperplano hasta el punto más cercano de cada patrón [66]. El objetivo principal es maximizar el margen para poder clasificar correctamente los patrones dados, es decir, cuanto mayor sea el tamaño del margen, más correctamente clasifica los patrones [68]. Los "vectores de soporte" en el contexto de las SVM, son observaciones que se encuentran en el límite o borde entre las clases en el espacio de características [67].

Ilustración 8

Dos clases separadas por un hiperplano en tres dimensiones



Nota. Extraído de [68].

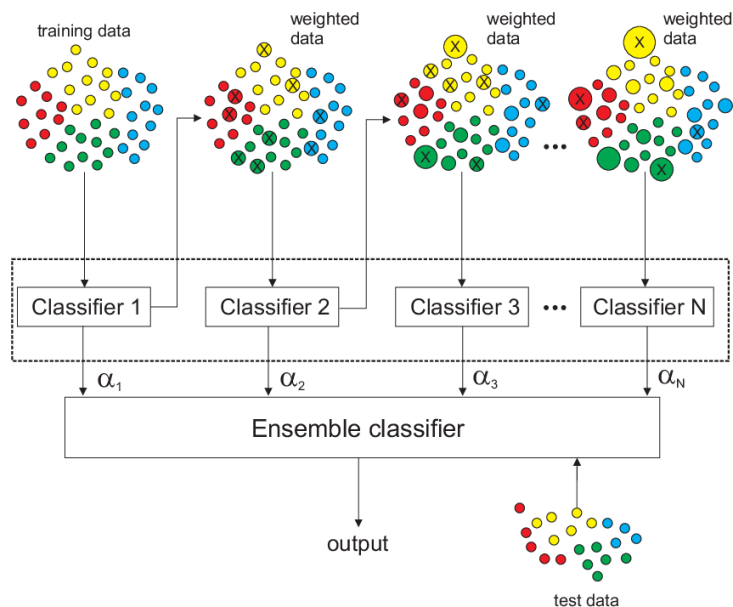
2.3.1.2 AdaBoost

Según [69], AdaBoost es un algoritmo de aprendizaje que busca mejorar el rendimiento de los algoritmos de clasificación, combina las predicciones de varios modelos base para generar una mejor predicción (o meta aprendizaje) que construye un clasificador de manera iterativa. En cada iteración, llama a un algoritmo de aprendizaje simple que devuelve un clasificador y le asigna un coeficiente de peso. La clasificación final se decide a través de un "voto" ponderado de los clasificadores base. Cuanto menor sea el error del clasificador base, mayor será su peso en el voto final.

Entre las ventajas que ofrece este algoritmo, se tiene que es flexible, ya que puede combinarse fácilmente con otros algoritmos de aprendizaje [70] y tiene buena capacidad de generalización, dado que, en problemas reales, es posible construir composiciones que son superiores en calidad a las de otros algoritmos de aprendizaje simples [71].

Ilustración 9

Diagrama del algoritmo AdaBoost para problemas de clasificación multiclase



Nota. Extraído de [111].

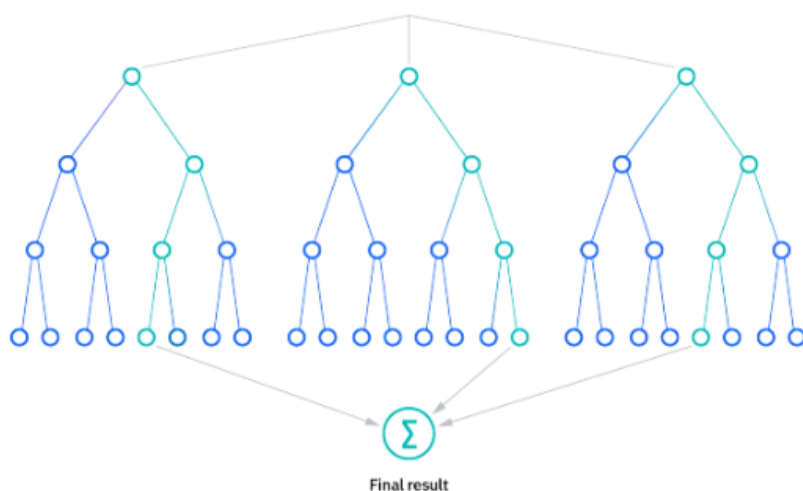
2.3.1.3 Random Forests

Random Forest es un clasificador compuesto por múltiples clasificadores basados en árboles [72]. Cada árbol vota por la clase más probable para una entrada dada, y operan de manera independiente. La aleatoriedad proviene del uso de vectores aleatorios distribuidos de manera independiente en la construcción de estos árboles [72]. La elección de utilizar el modelo de Bosques Aleatorios se justifica por dos razones clave: (1) como el Bosque Aleatorio puede generar resultados de predicción a partir de todos sus componentes base en lugar de depender de un solo puntaje de probabilidad; (2) es fácil obtener la importancia de las características en cada uno de los componentes base. La primera razón posibilita la construcción de una Red Neuronal Profunda (DNN) posterior al detector de características, algo que no se lograría si el detector solo proporcionara una única predicción, como sucede en las SVM. La segunda razón simplifica el

proceso de evaluación de características para el modelo completo, a diferencia de otros clasificadores como los métodos basados en kernel, que pueden no incorporar de manera natural un mecanismo de selección de características. [73]

Ilustración 10

Representación de Random Forest



Nota. Extraído de [105].

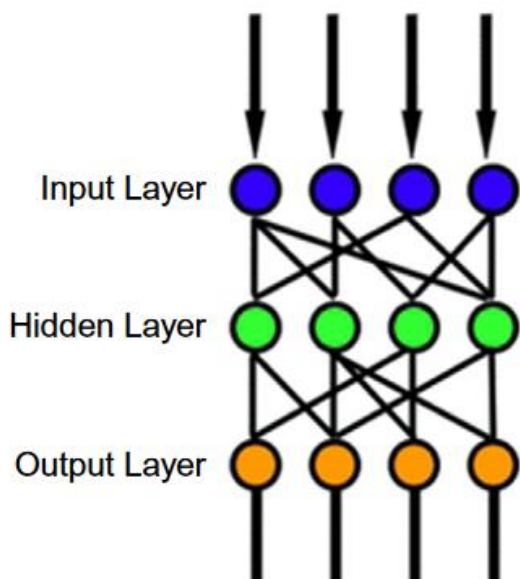
2.3.1.4 MLP

La estructura básica de una red neuronal de perceptrones multicapa incluye una capa de entrada, una o varias capas ocultas y una capa de salida. De acuerdo a [74], la capa de entrada contiene tantas neuronas como variables independientes, siendo igual a la cantidad de indicadores aplicados. De manera similar, la capa de salida tiene tantas neuronas como variables independientes, representando los subperiodos propuestos. La estructura de la capa oculta depende del tipo, tamaño y velocidad de ejecución del programa para resolver el problema [74]. Las redes neuronales de perceptrones multicapa (MLP) están diseñadas para aproximar cualquier función

continua y pueden resolver problemas que no son linealmente separables. Los principales casos de uso de MLP incluyen clasificación de patrones, reconocimiento, predicción y aproximación. [75]

Ilustración 11

Representación esquemática de una MLP con una capa oculta



Nota. Extraído de [75].

2.3.2 Pruebas en Modelos de Aprendizaje Profundo

Tras las pruebas con los modelos de aprendizaje poco profundo, se puso a prueba modelos de aprendizaje profundo para revisar el rendimiento que tendrían estos en distintos experimentos y finalizar con la selección del modelo con mejor rendimiento.

2.3.2.1 Pre-procesamiento

Se cargan las imágenes y sus etiquetas de los archivos en formato TFRecord. Con el objetivo de abordar el desbalance entre las imágenes correspondientes a sujetos con EP y sujetos sin EP, se implementó aumento de datos, específicamente para el conjunto de imágenes de sujetos sanos (HC) y aquellos con EP predominante en el lado izquierdo. Este proceso de aumento de

datos involucró la aplicación de la normalización ZCA whitening y rotación de 90 grados. Como resultado, se logró reducir la disparidad inicial de 360 registros a una diferencia de 200 registros, mitigando de manera efectiva el desbalance entre las clases y mejorando la representación del conjunto de datos. Finalmente, con el desbalance mitigado, las imágenes y sus etiquetas se agrupan y dividen en conjuntos de entrenamiento (70%) y prueba (30%).

2.3.2.2 Experimentación en Modelos

Se utilizarán varios modelos de redes neuronales profundas para la tarea de procesamiento de imágenes. Estos modelos incluyen las Redes Neuronales Convolucionales (CNN), en particular, los modelos **VGG16**, **VGG19**, **ResNet**, **MobileNet** e **Inception**. Además, como alternativa a las CNN, se empleará el modelo **Vision Transformer (ViT)**.

La experimentación en modelos de redes neuronales convolucionales se basó en el aprendizaje por transferencia utilizando los modelos **VGG16**, **VGG19**, **ResNet**, **MobileNet** e **Inception**. Cada experimento incluyó un aumento de datos mediante rotación, desplazamiento, cizallamiento, zoom y volteo horizontal. El optimizador utilizado fue Adam, con una función de pérdida de entropía cruzada binaria, y se midieron métricas de precisión. Los modelos se entrenaron a lo largo de 100 épocas y se evaluaron los resultados que se encuentran detallados en las siguientes figuras del próximo capítulo: **Tabla 17** (Resultados en modelos de aprendizaje profundo), **Ilustración 25** (Progreso de precisión y pérdida en el mejor modelo) e **Ilustración 26** (Matriz de confusión del mejor modelo).

Para cada experimento con las arquitecturas VGG, se cargó el modelo correspondiente (VGG16 o VGG19) y se ajustaron ciertas capas para adaptarse al problema específico. En el primer experimento, se congelaron las capas convolucionales, se aplicaron capas de normalización y se añadieron capas completamente conectadas. En el segundo, se ajustaron las últimas capas

convolucionales, y en el tercero, se modificaron aún más capas convolucionales con la adición de regularización, normalización de lotes y capas de desconexión (dropout). En los experimentos con MobileNet, se ajustaron más capas intermedias, utilizando tanto técnicas de normalización y capas densas en los primeros dos experimentos como una combinación de ajuste de capas convolucionales específicas y técnicas de regularización en el tercer experimento. Por otro lado, en los experimentos con ResNet, se observó un enfoque en ajustar capas convolucionales en los extremos de la red, ya sea congelando la mayoría de las capas y ajustando solo las últimas completamente conectadas, o ajustando capas específicas en las etapas iniciales con técnicas de regularización. Por último, en los experimentos con Inception, el primer experimento se asemeja al de los modelos VGG y ResNet. En el segundo experimento, se empleó el promedio global en lugar de aplanar las características como en el segundo experimento con la ResNet y en el tercer experimento, se aplicaron técnicas de regularización, normalización de lotes y capas de abandono (dropout) en las últimas capas del modelo.

Se exploró adicionalmente el modelo Vision Transformer (ViT). Se definieron capas personalizadas para extraer y codificar parches de las imágenes. Se utilizó un generador de datos para aplicar aumentos como rotación y desplazamiento del mismo modo en que se realizó con los modelos CNN. Después de entrenar el modelo con datos aumentados, se evaluó en un conjunto de prueba y se mostró la precisión. La implementación incluyó visualizaciones de las curvas de pérdida y precisión, así como la matriz de confusión.

2.3.2.3 *Redes Neuronales Convulcionales – CNN*

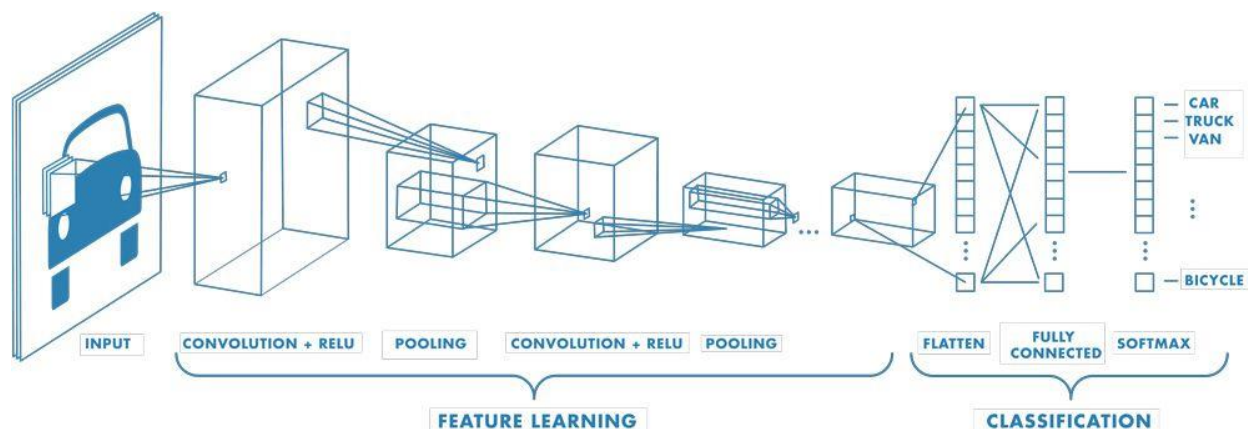
Las CNN constituyen una categoría especial de redes neuronales profundas diseñadas para tareas específicas de clasificación de imágenes, ya que son capaces de procesar representaciones

matriciales de datos de entrada. Los enfoques más avanzados de las CNN se centran en su utilización como CNN bidimensionales (2D) [79].

Las CNN constan de cuatro capas: la capa de convolución, la capa de reducción, la capa completamente conectada y la capa de salida [18]. Estas cuatro capas se ilustran en Ilustración 13. La capa de convolución utiliza filtros para realizar la convolución de las imágenes de entrada, extrayendo las características fundamentales. El filtro tiene el mismo tamaño, pero un valor constante más pequeño que la imagen de entrada [20]. La capa combinará dos capas convolucionales consecutivas. Esto reduce la cantidad de parámetros y la carga computacional al realizar representaciones de muestreo descendente. La función en la capa de agrupación puede generar un valor maximizado o promediado [82]. La tercera capa es la capa completamente conectada que se asemeja a una red neuronal MLP. Por lo general, se encuentra en la capa clasificadora. Recibe la entrada de la capa de salida convolucional, que se aplana antes de ser enviada a la capa siguiente. La distribución uniforme de la salida implica desenrollar todos los valores obtenidos después de la última capa convolucional en un vector (matriz tridimensional) [83]. Una función de activación desempeña un papel fundamental en las capas de convolución (la capa de no linealidad). La salida filtrada proporciona otra función matemática conocida como activación [83]. ReLU, abreviatura de la Unidad Lineal Rectificada (Rectified Linear Unit, en inglés) [84], es la función de activación más común en la extracción de características mediante el uso de CNN.

Ilustración 12

Capas de Red Neuronal Convolutiva



Nota. Extraído de [87].

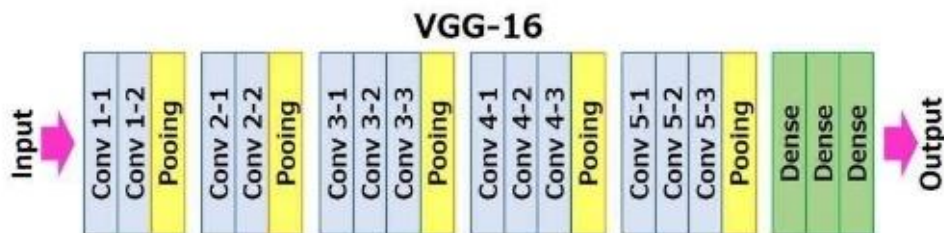
En las pruebas realizadas con CNN, se empleó la técnica de Transfer Learning con modelos preentrenados, los cuales fueron VGG16, VGG19, ResNet, MobileNet e Inception. Estos cuatro modelos tienen sus respectivas implementaciones en la librería keras.

2.3.2.3.1 VGG16

VGG16 consta de 13 capas convolucionales, seguidas por 3 capas completamente conectadas. Obtuvo un rendimiento de vanguardia en el conjunto de datos ImageNet, que es un referente ampliamente utilizado para tareas de clasificación de imágenes. El tamaño de la entrada de imagen de la red es de 224 por 224. [85]. Las ventajas del uso de la VGG16 están relacionadas a su profundidad que posee pocas capas convolucionales y que permite el uso de transfer learning para poder preentrenar con un conjunto de datos similares para poder acelerar el entrenamiento y el rendimiento.[97]

Ilustración 13

Arquitectura de la VGG-16



Nota. Extraído de [101].

Tabla 1

Hiperparámetros de la VGG16

Arquitectura	Hiperparámetro	Descripción
VGG16	Capas congeladas	Número de capas iniciales de la red que se mantienen no entrenables, generalmente para la transferencia de aprendizaje.
	Tamaño del kernel	Tamaño de la ventana convolucional utilizada para extraer características de la imagen.
	Función de activación	La función matemática aplicada a la salida de una neurona.
	Padding	La forma en que se manejan los bordes de la imagen durante la convolución.
	Regularización L2	Tasa de regularización que penaliza los pesos grandes en la función de pérdida para prevenir el sobreajuste.

Tabla 2*Experimentos con la VGG16*

Número de Experimento	Arquitectura de VGG16	Descripción
Experimento 1	Arquitectura personalizada 1	Utiliza una arquitectura personalizada que extrae características de las capas 'block3_conv3', 'block4_conv3' y 'block5_conv3' de VGG16. Se aplican capas de normalización y se agregan capas totalmente conectadas.
Experimento 2	Arquitectura personalizada 2	Utiliza la arquitectura VGG16 con las capas completamente conectadas originales ('include_top=True') y congela las primeras tres capas convolucionales. Se agrega una capa de salida para la clasificación binaria.
Experimento 3	Arquitectura personalizada 3	Utiliza una arquitectura personalizada que congela las primeras 11 capas convolucionales de VGG16 ('include_top=False'). Se agregan capas de convolución y normalización, y se aplican tasas de dropout en las capas convolucionales y totalmente conectadas.

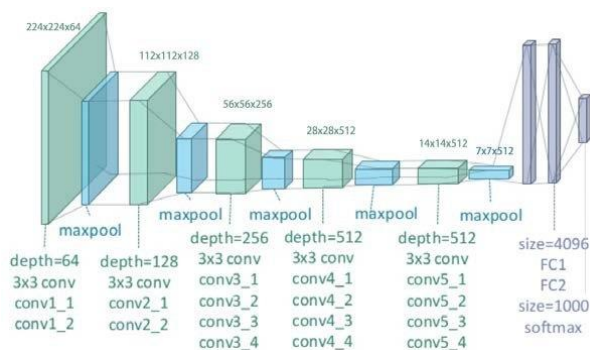
2.3.2.3.2 VGG19

VGG-19 es una red neuronal convolucional que consta de 19 capas en su profundidad. Tiene la capacidad de cargar una versión preentrenada de la red, la cual fue entrenada con más de un millón de imágenes procedentes de la base de datos de ImageNet. El tamaño de la entrada de imagen de la red es de 224 por 224 al igual que en VGG16 [86]. La VGG16 al tener cinco bloques

de capas convolucionales, permitirá descubrir de forma autónoma en las imágenes las características de esta [98].

Ilustración 14

Arquitectura de la VGG-19



Nota. Extraído de [102].

Tabla 3

Experimentos con la VGG19

Número de Experimento	Arquitectura de VGG19	Descripción
Experimento 1	Arquitectura personalizada 1	Utiliza una arquitectura personalizada que extrae características de las capas 'block3_conv3', 'block4_conv3' y 'block5_conv3' de VGG19. Se aplican capas de normalización y se agregan capas totalmente conectadas.
Experimento 2	Arquitectura pre-entrenada 1	Utiliza la arquitectura VGG19 con las capas completamente conectadas originales ('include_top=True') y congela las primeras tres capas

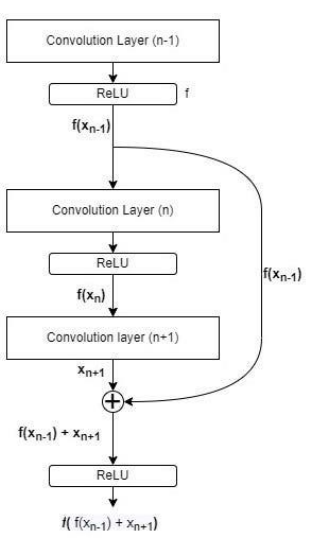
		convolucionales. Se agrega una capa de salida para la clasificación binaria.
Experimento 3	Arquitectura personalizada 2	Utiliza una arquitectura personalizada que congela las primeras 11 capas convolucionales de VGG19 ('include_top=False'). Se agregan capas de convolución y normalización, y se aplican tasas de dropout en las capas convolucionales y totalmente conectadas.

2.3.2.3.3 ResNet

ResNet se caracteriza por introducir la idea de conexiones residuales. Esto permite que el gradiente fluya más fácilmente a través de la red, lo que facilita el entrenamiento de redes profundas. Estas redes son conocidas por su capacidad para alcanzar un alto rendimiento en la clasificación de imágenes y la detección de objetos.

Ilustración 15

Funcionamiento de una ResNet



Nota. Extraído de [103].

Tabla 4*Hiperparámetros de la ResNet*

Arquitectura	Hiperparámetro	Descripción
ResNet	Tamaño del filtro (kernel size)	El tamaño del filtro convolucional utilizado.
	Número de capas residuales	La profundidad de la red, que puede variar.
	Tasa de aprendizaje (learning rate)	La velocidad de ajuste de los pesos durante el entrenamiento.
	Tamaño del batch (batch size)	El número de ejemplos de entrenamiento en cada lote.

Tabla 5*Experimentos con la ResNet*

Número de Experimento	Arquitectura de ResNet	Descripción
Experimento 1	Arquitectura personalizada 1	Utiliza una arquitectura personalizada que extrae características de las capas 'conv3_block4_out', 'conv4_block6_out' y 'conv5_block3_out' de ResNet50V2. Se aplican capas de normalización y se agregan capas totalmente conectadas.
Experimento 2	Arquitectura pre-entrenada 1	Utiliza la arquitectura ResNet50V2 con las capas completamente conectadas originales ('include_top=True') y congela las primeras tres capas

convolucionales. Se agrega una capa de salida para la clasificación binaria.

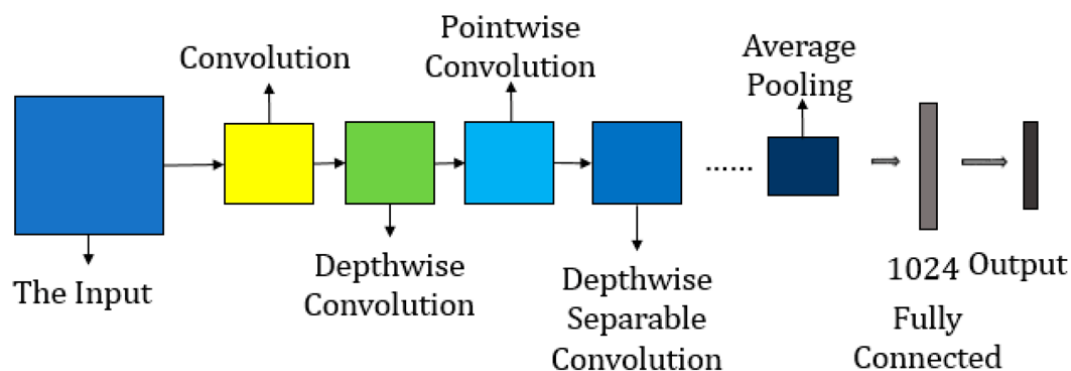
Experimento 3 Arquitectura Utiliza una arquitectura personalizada que congela todas personalizadas 2 las capas excepto las últimas 10. Se aplican capas de convolución, normalización y dropout en las capas convolucionales y totalmente conectadas.

2.3.2.3.4 MobileNet

MobileNet está diseñada para aplicaciones en dispositivos móviles y embebidos con restricciones computacionales. La flexibilidad de este modelo radica en la capacidad de ajustar el ancho y la profundidad de la red, lo que proporciona un equilibrio entre eficiencia y precisión. Su característica única es su uso de convoluciones separables en profundidad, que pueden considerarse como convoluciones estándar divididas en convoluciones en profundidad y convoluciones 1 por 1 en punto. [94]

Ilustración 16

Arquitectura de una MobileNet



Nota. Extraído de [100].

Tabla 6*Hiperparámetro de la MobileNet*

Arquitectura	Hiperparámetro	Descripción
MobileNet	Ancho de ancho (width multiplier)	Controla el tamaño de la red y la eficiencia computacional.
	Resolución de entrada	El tamaño de la imagen de entrada.

Tabla 7*Experimentos con MobileNet*

Número de Experimento	Arquitectura de	Descripción
	MobileNetV3 Small	
Experimento 1	Arquitectura personalizada 1	Utiliza una arquitectura personalizada que extrae características de la última capa convolucional y aplica capas de normalización y capas totalmente conectadas.
Experimento 2	Arquitectura pre-entrenada 1	Utiliza la arquitectura MobileNetV3 Small pre-entrenada con las capas completamente conectadas originales ('include_top=True') y congela el modelo base. Se agrega una capa de salida para la clasificación binaria.
Experimento 3	Arquitectura personalizada 2	Utiliza una arquitectura personalizada que congela todas las capas excepto las últimas 20. Se extraen características de ciertas capas y se aplican capas de

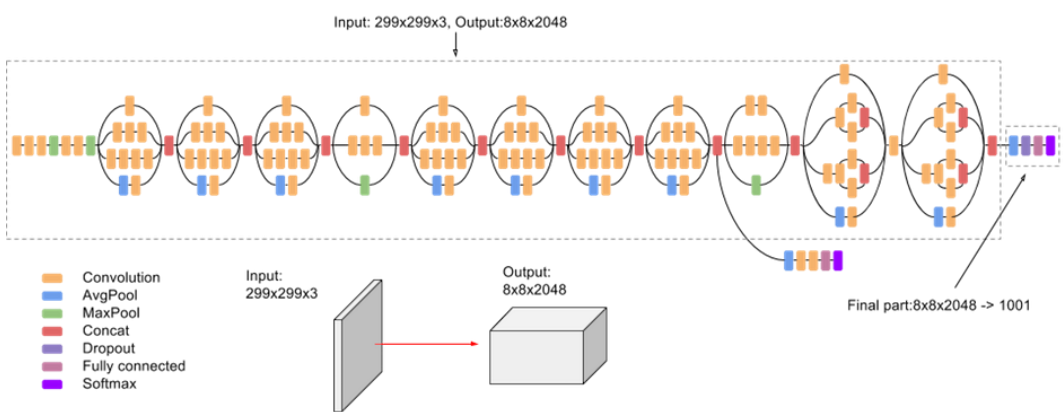
convolución, normalización y dropout. Se agregan capas totalmente conectadas para la clasificación.

2.3.2.3.5 Inception

Inception es una arquitectura que se destaca por el enfoque que tiene en la precisión en tareas de visión por computadoras. La Inception v3 es una red convolucional que cuenta con 48 capas de profundidad, que usa bloques de construcción simétricos y asimétricos, incluyendo una variedad de operaciones como convoluciones, promedios y máximos de agrupación, concatenaciones, eliminaciones y capas completamente conectadas [95]. Entre sus ventajas están su eficiencia computacional, la baja cantidad de parámetros y su capacidad de escalar redes de forma efectiva [96].

Ilustración 17

Arquitectura de una Inception



Nota. Extraído de [95]

Tabla 8*Hiperparámetros de la Inception*

Arquitectura	Hiperparámetro	Descripción
Inception	Tamaño del filtro (kernel size)	El tamaño del filtro convolucional utilizado.
	Número de módulos Inception	La cantidad de módulos Inception en la red.
	Tasa de aprendizaje (learning rate)	La velocidad de ajuste de los pesos durante el entrenamiento.

Tabla 9*Experimentos con la Inception*

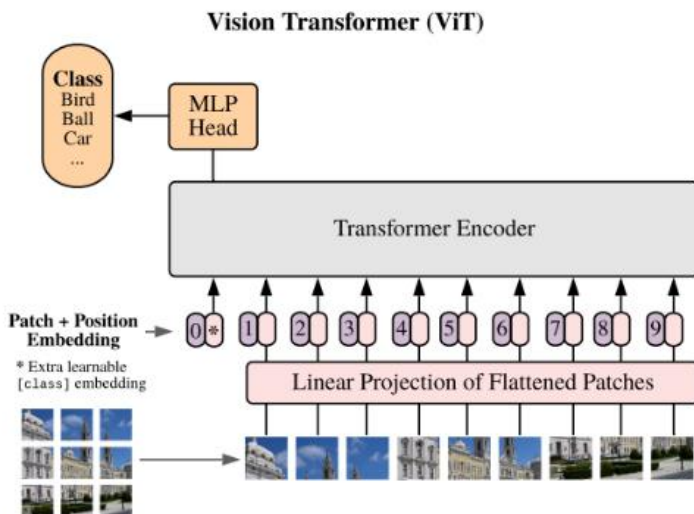
Número de Experimento	Arquitectura de Inception	Descripción
Experimento 1	Arquitectura personalizada 1	Utiliza una arquitectura personalizada que extrae características de ciertas capas de InceptionV3, aplica capas de normalización y capas totalmente conectadas.
Experimento 2	Arquitectura pre-entrenada 1	Utiliza la arquitectura InceptionV3 pre-entrenada con las capas completamente conectadas originales ('include_top=True') y congela el modelo base. Se agrega una capa de salida para la clasificación binaria.
Experimento 3	Arquitectura personalizada 2	Utiliza una arquitectura personalizada que congela las primeras 11 capas de InceptionV3, extrae características de ciertas capas, aplica capas de normalización, convolución y dropout. Se agregan capas totalmente conectadas para la clasificación.

2.3.2.4 Vision Transformer

La arquitectura Vision Transformer (ViT) representa una alternativa a las Convolutional Neural Networks (CNN) en el campo de la visión por computadora. En lugar de emplear capas convolucionales como las CNN, las ViT descomponen las imágenes en parches y los transforman en vectores, de modo que pueden ser procesados como palabras en un transformer convencional. Entre las ventajas notables de las ViT se encuentra la utilización de bloques de atención que facilitan la detección de relaciones entre diversos elementos dentro de la imagen [99].

Ilustración 18

Arquitectura de una ViT



Nota. Extraído de [104].

Tabla 10*Hiperparámetros de ViT*

Arquitectura	Hiperparámetro	Descripción
ViT	Tamaño del batch (batch size)	El número de parches de la imagen en cada lote.
	Número de cabezas de atención	La cantidad de cabezas de atención en el modelo.
	Dimensiones de los embeddings	Las dimensiones de los vectores de embeddings.

Tabla 11*Experimentos con ViT*

Número de Experimento	Arquitectura	Descripción
	InceptionV3	
Experimento 1	Arquitectura personalizada	Utiliza una arquitectura personalizada de Vision Transformer (ViT) que extrae parches de la imagen, los codifica, aplica atención multi-cabeza y capas MLP para la clasificación binaria.

2.4 Desarrollo de Prototipo**2.4.1 Actores del Sistema**

Gracias a las reuniones tanto virtuales como presenciales con el cliente, logramos identificar los siguientes roles dentro del sistema:

2.4.1.1 Actor Doctor

Este rol está diseñado para médicos que utilizarán el sistema para realizar diagnósticos de enfermedades. El sistema ofrece las siguientes funcionalidades:

- Cargar archivos que contengan la escritura del paciente a ser diagnosticado.
- Visualizar los resultados del diagnóstico.
- Registrar observaciones pertinentes al diagnóstico proporcionado por el sistema.

2.4.1.2 Actor Administrador

Este rol se diseñó para el administrador del sistema, cuya responsabilidad es supervisar los diagnósticos y asegurar el funcionamiento del modelo basado en inteligencia artificial. Entre sus funciones se incluyen:

- Crear, editar y eliminar usuarios.
- Visualizar la información de los diagnósticos realizados por los médicos.
- Acceder a la ubicación de los archivos con la escritura de los pacientes.

2.4.2 Requerimientos no funcionales

- **Entorno:**

La interacción de los médicos con el sistema debe ser a través de un computador con conexión a internet.

- **Implementación:**

El sistema deberá ser implementado en Python en el framework de FastApi

- **Escalabilidad y Almacenamiento**

El sistema debe ser capaz de almacenar de manera eficiente y segura un volumen medio de archivos de escritura de pacientes, junto con sus respectivos diagnósticos.

2.4.3 Alcance de la solución

La solución será una página de acceso exclusivo para médicos autorizados que le ayudará a calcular la probabilidad de que un paciente tenga o no EP a partir de datos extraídos de una tableta que se utilizó para distintas pruebas de escritura. Para el alcance de este proyecto, se utilizará el lenguaje de programación Python para el desarrollo de los modelos, la librería React y el marco web FastAPI para el desarrollo de la aplicación web en Front-End y Back-End respectivamente.

2.4.4 Riesgos y beneficios del prototipo de la solución

Riesgos:

- **Errores de lectura de archivos.**

Que no se entienda bien el tipo de formato de archivo de escritura que se tiene que subir a la aplicación.

Beneficios:

- **Accesibilidad**

Es fácilmente accesible la recopilación de datos de pacientes al ser la escritura una actividad cotidiana para la mayoría de las personas.

- **Poco invasivo**

La escritura no representa una amenaza para la salud del paciente y es cómoda de realizar.

- **Bajo costo**

No es necesario el uso de equipos médicos costosos ni procedimientos complicados. Solo es necesario tener un dispositivo para poder registrar la escritura, haciéndolo económicamente viable.

- **Monitoreo continuo**

La escritura puede ser monitoreada de manera continua a lo largo del tiempo, lo que permite un seguimiento más preciso de la evolución de la enfermedad.

2.4.5 Arquitectura del prototipo

2.4.5.1 Tecnologías

Esta plataforma web estará integrada por las siguientes tecnologías para el front-end y el back-end:

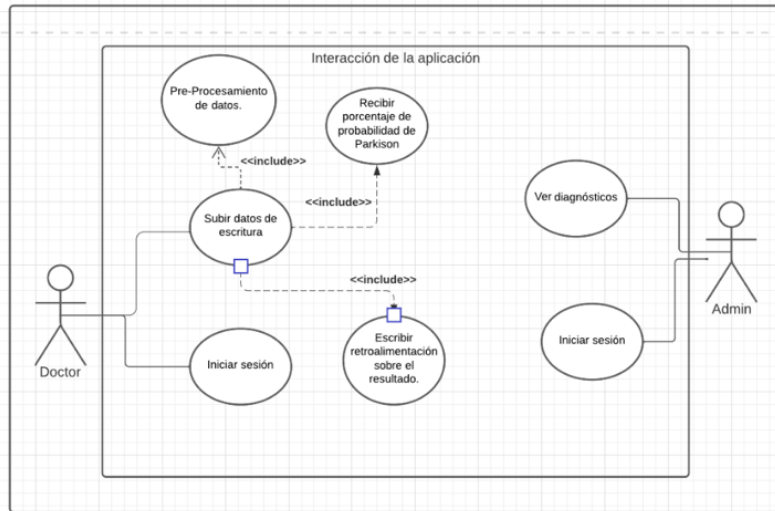
- **FastAPI:** Este framework de Python facilita el desarrollo del backend, para realizar los endpoints de la aplicación de manera rápida.
- **React:** Este framework de JavaScript nos brinda las herramientas necesarias para desarrollar el frontend de la aplicación de manera eficaz y efectiva.
- **MySQL:** Sistema de gestión de base de datos relacional que proporciona una estructura de base de datos robusta y escalable que puede gestionar eficientemente grandes cantidades de datos. Permite realizar consultas SQL para acceder y manipular la información almacenada en la base de datos.

2.4.5.2 Diagramas de la arquitectura

En esta sección se muestra los requerimientos funcionales de la plataforma. Los usuarios de esta plataforma van a ser el: Doctor y administrador. El doctor es el usuario principal, en cual va a existir la opción de subir datos de escritura de los sujetos, a través de archivos sbs (los cuales tienen que ser preprocesados). En cambio, el administrador se va a encargar de ver los diagnósticos y todos los archivos subidos al sistema.

Ilustración 19

Diagrama de casos de uso de la aplicación

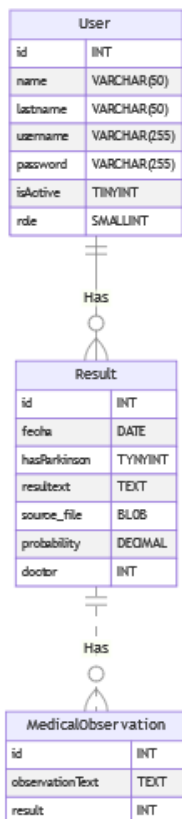


Este sistema contiene las siguientes clases:

- **User:** Tiene un nombre, apellido, id, username, contraseña, indicador de si el usuario sigue activo y un rol que puede ser el de doctor (0) o administrador (1).
- **Diagnosis (Diagnóstico):** Tiene su id, la fecha en que se realizó, la ubicación del archivo correspondiente al diagnóstico, la probabilidad calculada y el id del médico para el que se realizó el diagnóstico.
- **MedicalObsevation:** Tiene su id, contenido de la observación, fecha en la que se realizó y el id del diagnóstico al que se asigna.

Ilustración 20

Diagrama de entidad-relación



2.4.6 Prototipo Actual

En esta sección, se muestran imágenes del prototipo actual de la aplicación web que fue diseñado en la aplicación 'Figma'. El prototipo se divide para los 2 roles que hay para satisfacer que el rol del médico y otra enfocada en las del rol del administrador. Para ambos roles, se tiene una pantalla de inicio de sesión (**Ilustración 21**).

Ilustración 21

Pantalla de Inicio de Sesión



The screenshot shows a login interface with a blue header bar containing the text "Iniciar Sesión". Below the header, there are two input fields: "Nombre de Usuario" and "Ingrese su contraseña". Below these fields is a blue button labeled "Inicia Sesión". Underneath the button is a link that says "¿No tienes una cuenta?, regístrate aquí". At the bottom of the form is another blue button labeled "Regístrate".

El rol de médico tiene la posibilidad de registrarse (**Ilustración 22**). Este mismo rol, tras iniciar sesión, tiene la posibilidad de subir los datos de la muestra de escritura para los que desee que se le genere una predicción (**Ilustración 23**). Una vez dado el resultado de la predicción, se da la opción al médico para agregar observaciones.

Ilustración 22

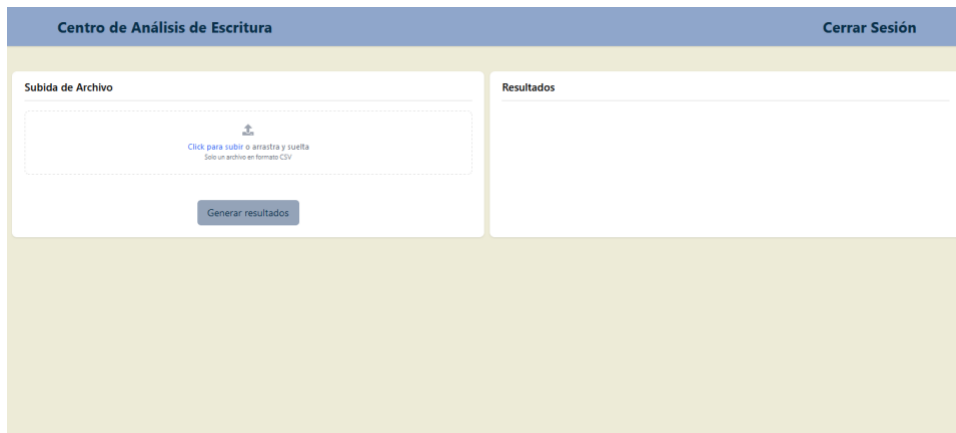
Pantalla de registro de médico



The screenshot shows a registration form for a doctor with a blue header bar containing the text "Nuevo Doctor". The form contains five input fields: "Nombre", "Apellido", "Nombre de Usuario", "Contraseña", and "Confirma Contraseña". Below these fields is a blue button labeled "Registro".

Ilustración 23

Pantalla principal para el médico



Para el rol del administrador, en su pantalla de inicio, se da opción para gestionar datos de los usuarios y de los resultados (**Ilustración 24**). Se ofrece las opciones de editar, crear e inhabilitar usuarios, y también la posibilidad de visualizar los datos de los resultados.

Ilustración 24

Pantalla principal para el administrador



CAPÍTULO 3

3. Resultados

En este capítulo se presentan los resultados de las pruebas con los modelos de aprendizaje poco profundo y aprendizaje profundo a los que se hace referencia en el **Capítulo 2**.

3.1 Experimentos Realizados

3.1.1 Entorno de Desarrollo

Se empleó Google Colab Pro [111] como entorno en la nube para ejecutar el código y llevar a cabo pruebas en modelos de aprendizaje profundo. Este servicio proporcionó recursos computacionales adicionales en comparación con la versión gratuita de Google Colab, lo que resultó crucial para el entrenamiento eficiente de modelos de aprendizaje profundo y la experimentación con diversas arquitecturas. Además, el uso de Colab Pro aceleró significativamente el tiempo de entrenamiento, permitiendo realizar iteraciones más rápidas en el desarrollo de los modelos. Para los modelos de aprendizaje poco profundo, se empleó la herramienta ya mencionada, pero también se trabajó en un entorno virtual con la ayuda de Anaconda [112]. El código utilizó varias bibliotecas en Python [113] para realizar tareas específicas. La librería TensorFlow [114] fue esencial para operaciones de aprendizaje profundo, manipulación de datos y procesamiento de imágenes. OpenCV [115] se utilizó para tareas de visión por computadora, especialmente el procesamiento de imágenes. Scikit-learn [116] fue utilizado de manera predominante en las pruebas con modelos de aprendizaje poco profundo y para obtener la matriz de confusión en los diferentes experimentos. Estas bibliotecas ofrecen funcionalidades para desarrollar y entrenar modelos predictivos y para procesar y analizar datos.

3.2 Comparación de modelos

Para el análisis de los modelos, se desarrolló una serie de experimentos utilizando diferentes arquitecturas para aprendizaje poco profundo y aprendizaje profundo. Estos experimentos se separaron de acuerdo con el tipo de clasificación para los modelos de aprendizaje poco profundo sí se consideró la clasificación multiclase; mientras que, para los modelos de aprendizaje profundo, los experimentos fueron diseñados para clasificación binaria.

3.2.1 Modelos de Aprendizaje Poco Profundo

Para los experimentos realizados con estos modelos, se consideraron tres conjuntos de datos que difieren en su forma de ser obtenidos (ver **Tabla 12**).

Tabla 12

Diferentes conjuntos de datos para modelos de aprendizaje poco profundo

Tipo de datos	Cantidad
Datos Médicos	88 características entre las que se encuentran características cinemáticas y estadísticas.
Datos Matemáticos	Un total de 761 características obtenidas usando datos de series de tiempo junto a la biblioteca “tsfresh”.
Datos Combinados	849 características que reúnen las características de los tipos anteriores.

Las **Tablas 13** a la **16** muestran los resultados de los experimentos realizados con los modelos de aprendizaje poco profundo, es importante señalar que el método de extracción de

características RFE (Recursive Feature Elimination) produjo mejores resultados de precisión al compararse con MW (Prueba U de Mann-Whitney) en la clasificación binaria y KW (Prueba de Kruskal-Wallis) en la clasificación multiclase para la mayoría de los casos y considerando el número de características en cada comparación.

Respecto a los valores más altos de precisión alcanzados por cada modelo bajo la clasificación binaria, AdaBoost y Random Forest utilizaron subconjuntos de datos extraídos con RFE para producir sus mejores valores de precisión que fueron 75% y 79.66% respectivamente, mientras que SVM y MLP emplearon los conjuntos totales de datos combinados y datos matemáticos para alcanzar 88.06% y 78.42% de precisión respectivamente. En cuanto a la clasificación multiclase, la mejor precisión con AdaBoost fue del 64.55% sin utilizar un método de extracción de atributos, mientras que Random Forest, SVM y MLP ocuparon subconjuntos generados con RFE para obtener sus mejores precisiones que fueron 74.03%, 74.25% y 71.46% respectivamente. Otro aspecto por señalar es el modo en que se repiten los resultados al utilizar MW y KW con los datos matemáticos y combinados, esto se puede evidenciar con AdaBoost empleando los subconjuntos con 20, 15 y 10 características extraídos con KW. Esto último nos permite afirmar que, para este contexto experimental, los métodos MW y KW producen subconjuntos de datos que pueden llegar a ser muy similares o incluso iguales en su contenido, lo cual genera resultados de gran similitud.

Tabla 13*Resultados de modelo AdaBoost para clasificación binaria y multiclase respectivamente*

Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción	Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción
AdaBoost	68,09%	Médicos	88	Ninguno	AdaBoost	53,17%	Médicos	88	Ninguno
	66,60%		25	RFE		56,71%		25	RFE
	66,60%		25	MW		55,78%		25	KW
	64,18%		20	RFE		54,47%		20	RFE
	65,85%		20	MW		53,17%		20	KW
	64,55%		15	RFE		52,80%		15	RFE
	65,85%		15	MW		52,24%		15	KW
	66,97%		10	RFE		52,80%		10	RFE
	65,30%		10	MW		50,37%		10	KW
AdaBoost	72,57%	Matemáticos	761	Ninguno	AdaBoost	60,44%	Matemáticos	761	Ninguno
	72,95%		25	RFE		64,36%		25	RFE
	66,97%		25	MW		54,29%		25	KW
	75,00%		20	RFE		58,21%		20	RFE
	67,16%		20	MW		55,04%		20	KW
	73,69%		15	RFE		59,14%		15	RFE
	67,72%		15	MW		52,98%		15	KW
	74,44%		10	RFE		60,63%		10	RFE
	66,60%		10	MW		51,50%		10	KW
AdaBoost	73,88%	Combinados	849	Ninguno	AdaBoost	64,55%	Combinados	849	Ninguno
	74,07%		25	RFE		63,25%		25	RFE
	66,97%		25	MW		55,97%		25	KW
	70,90%		20	RFE		61,57%		20	RFE
	67,16%		20	MW		55,04%		20	KW
	73,51%		15	RFE		61,57%		15	RFE
	67,72%		15	MW		52,98%		15	KW
	72,20%		10	RFE		60,07%		10	RFE
	66,60%		10	MW		51,50%		10	KW

Tabla 14*Resultados de modelo Random Forest para clasificación multiclase y binaria respectivamente*

Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción	Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción
Random Forest	60.59%	Médicos	88	Ninguno	Random Forest	70.78%	Médicos	88	Ninguno
	62.72%		25	RFE		72.57%		25	RFE
	57.04%		25	KW		68.99%		25	MW
	61.93%		20	RFE		72.76%		20	RFE
	58.32%		20	KW		68.77%		20	MW
	62.58%		15	RFE		72.50%		15	RFE
	56.55%		15	KW		68.17%		15	MW
	61.99%		10	RFE		72.84%		10	RFE
	54.42%		10	KW		68.69%		10	MW
Random Forest	72.87%	Matemáticos	761	Ninguno	Random Forest	78.92%	Matemáticos	761	Ninguno
	73.20%		25	RFE		78.58%		25	RFE
	61.07%		25	KW		71.87%		25	MW
	73.04%		20	RFE		78.13%		20	RFE
	60.55%		20	KW		70.30%		20	MW
	71.75%		15	RFE		77.20%		15	RFE
	56.30%		15	KW		68.40%		15	MW
	69.74%		10	RFE		77.09%		10	RFE
	56.45%		10	KW		68.32%		10	MW
Random Forest	73.02%	Combinados	849	Ninguno	Random Forest	79.63%	Combinados	849	Ninguno
	73.84%		25	RFE		79.66%		25	RFE
	61.31%		25	KW		71.87%		25	MW
	74.03%		20	RFE		79.44%		20	RFE
	60.55%		20	KW		70.30%		20	MW
	71.75%		15	RFE		78.54%		15	RFE
	56.30%		15	KW		68.40%		15	MW
	71.87%		10	RFE		76.01%		10	RFE
	56.45%		10	KW		68.32%		10	MW

Tabla 15*Resultados de modelo SVM para clasificación binaria y multiclase respectivamente*

Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción	Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción
SVM	72.39%	Médicos	88	Ninguno	SVM	57.46%	Médicos	88	Ninguno
	56.53%		25	RFE		66.60%		25	RFE
	70.14%		25	MW		59.89%		25	KW
	56.52%		20	RFE		63.80%		20	RFE
	71.26%		20	MW		60.82%		20	KW
	56.52%		15	RFE		64.73%		15	RFE
	69.02%		15	MW		59.51%		15	KW
	56.52%		10	RFE		62.12%		10	RFE
	68.84%		10	MW		57.08%		10	KW
SVM	76.50%	Matemáticos	761	Ninguno	SVM	61.23%	Matemáticos	761	Ninguno
	78.54%		25	RFE		74.25%		25	RFE
	72.20%		25	MW		62.12%		25	KW
	78.91%		20	RFE		72.57%		20	RFE
	71.26%		20	MW		60.82%		20	KW
	76.11%		15	RFE		72.57%		15	RFE
	69.40%		15	MW		56.15%		15	KW
	75.19%		10	RFE		65.29%		10	RFE
	69.59%		10	MW		57.84%		10	KW
SVM	88.06%	Combinados	849	Ninguno	SVM	60.07%	Combinados	849	Ninguno
	80.22%		25	RFE		72.39%		25	RFE
	72.20%		25	MW		61.57%		25	KW
	80.03%		20	RFE		71.64%		20	RFE
	71.26%		20	MW		60.82%		20	KW
	78.54%		15	RFE		68.65%		15	RFE
	70.00%		15	MW		56.15%		15	KW
	75.19%		10	RFE		66.97%		10	RFE
	69.59%		10	MW		57.84%		10	KW

Tabla 16*Resultados de MLP para clasificación binaria y multiclase respectivamente*

Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción	Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción
	73.00%		88	Ninguno		60.00%		88	Ninguno
	70.90%		25	RFE		59.89%		25	RFE
	69.58%		25	MW		59.88%		25	KW
	70.71%		20	RFE		60.63%		20	RFE
MLP	69.40%	Médicos	20	MW	MLP	61.01%	Médicos	20	KW
	68.28%		15	RFE		59.52%		15	RFE
	67.16%		15	MW		57.09%		15	KW
	68.65%		10	RFE		59.32%		10	RFE
	67.30%		10	MW		56.34%		10	KW
	78.42%		761	Ninguno		67.30%		761	Ninguno
	77.05%		25	RFE		71.46%		25	RFE
	70.35%		25	MW		59.70%		25	KW
	75.75%		20	RFE		66.79%		20	RFE
MLP	69.03%	Matemáticos	20	MW	MLP	59.32%	Matemáticos	20	KW
	73.14%		15	RFE		65.86%		15	RFE
	69.59%		15	MW		54.85%		15	KW
	73.88%		10	RFE		64.55%		10	RFE
	69.22%		10	MW		55.03%		10	KW
	78.10%		849	Ninguno		68.70%		849	Ninguno
	78.18%		25	RFE		67.91%		25	RFE
	71.27%		25	MW		59.70%		25	KW
	77.05%		20	RFE		65.30%		20	RFE
MLP	67.35%	Combinados	20	MW	MLP	58.76%	Combinados	20	KW
	76.87%		15	RFE		65.30%		15	RFE
	68.84%		15	MW		54.66%		15	KW
	72.95%		10	RFE		64.93%		10	RFE
	69.40%		10	MW		54.85%		10	KW

3.2.2 Modelos de Aprendizaje Profundo

Con modelos se realizaron tres tipos de experimentos, en el primero se varía el tamaño del lote o batch en el segundo, para el segundo experimento se cambió la tasa de aprendizaje y en el tercer tipo de experimento se experimentó con lotes de diferentes tamaños.

La **Tabla 17** muestra los resultados experimentales para cada modelo, y se pudo observar que el mejor resultado se dió utilizando el modelo preentrenado VGG19 con una precisión del 67.79% en el tercer experimento. Considerando los tres experimentos, el modelo VGG19 tuvo el mejor rendimiento en cuanto a precisión en promedio.

Tabla 17

Resultados en modelos de aprendizaje profundo

Modelo	Experimento	Tamaño Batch	Épocas	Tasa de Aprendizaje	Precisión en Prueba (%)
VGG16	1	32	100	0.0001	61.79
	2	64	100	0.0001	63.89
	3	32	100	1.00E-05	66.52
VGG19	1	32	100	0.0001	64.04
	2	64	100	0.0001	60.51
	3	32	100	1.00E-05	67.79
ResNet	1	32	100	0.0001	53.68
	2	64	100	0.0001	67.72
	3	32	100	1.00E-05	59.46
MobileNet	1	32	100	0.0001	55.26
	2	64	100	0.0001	57.43
	3	32	100	1.00E-05	59.53
Inception	1	32	100	0.0001	55.86
	2	64	100	0.0001	66.22
	3	32	100	1.00E-05	62.09

En relación con el modelo con mejor resultado, la **Ilustración 25** revela un patrón donde la precisión de entrenamiento exhibe un crecimiento acelerado, a la par que se intensifica la disparidad entre esta métrica y la precisión en el conjunto de validación. Esta tendencia evidencia un marcado caso de memorización de los datos de entrenamiento. Además, al igual que ocurre con las curvas de precisión, se observa un ensanchamiento significativo en la brecha entre las curvas de pérdida de entrenamiento y validación, señalando que el modelo tiende a memorizar los datos de entrenamiento en lugar de generalizar patrones de manera efectiva.

La matriz de confusión representada en la **Ilustración 26**, muestra que ha habido más aciertos que fallos en cuanto a las etiquetas (Parkinson / No Parkinson). Esto se debe a que hay 514 aciertos con ausencia de EP y los 389 aciertos con presencia de EP, con los 223 y 206 fallos respectivamente en la predicción. Estos fallos, aunque presentes, deben interpretarse en el contexto del balance general de aciertos, y podrían ser objeto de análisis adicional para identificar posibles patrones específicos de error y mejorar la capacidad predictiva del modelo en áreas específicas.

Ilustración 25

Progreso de precisión y pérdida en el mejor modelo

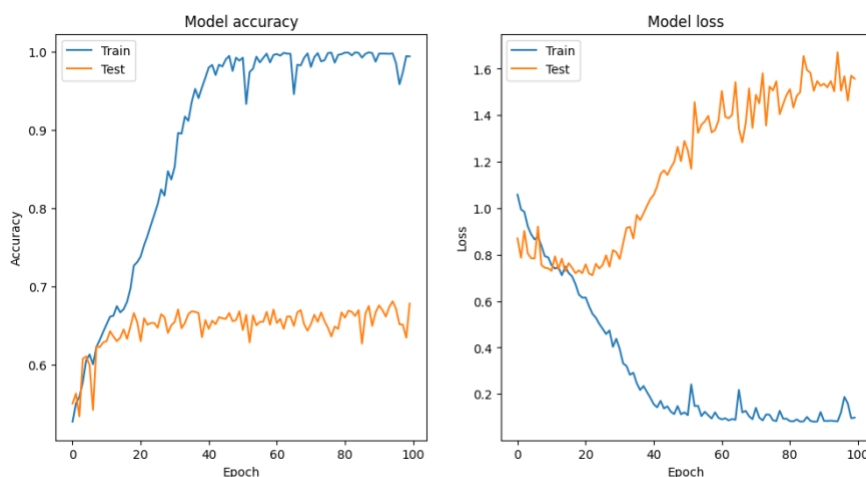
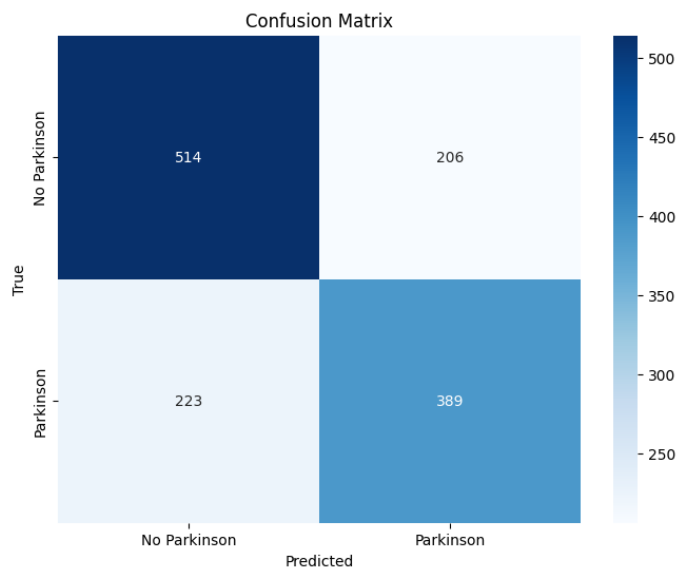


Ilustración 26

Matriz de confusión del mejor modelo



3.2.3 Comparación entre modelos y selección del mejor modelo

Al finalizar las pruebas con los modelos de aprendizaje poco profundo y los modelos de aprendizaje profundo, se compararon los resultados obtenidos de cada uno de estos. La comparación se enfocó principalmente en la precisión en el conjunto de pruebas. En primer lugar, el mejor modelo de aprendizaje profundo con su precisión del 67.79% en clasificación binaria no figura como un buen candidato para mejor modelo, así que la comparación se limita a los modelos de aprendizaje poco profundo. En lo que respecta a los modelos de aprendizaje poco profundo, en la **Tabla 18** se muestran los mejores resultados tanto para clasificación binaria como para clasificación multiclase. Dado que, para nuestro proyecto, se debía incorporar un modelo de clasificación multiclase.

Tabla 18

Resultados de mejores modelos para clasificación binaria y multiclase

Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción	Modelo	Precisión (Test)	Conjunto	# Atributos	Método Extracción
AdaBoost	75%	Matemáticos	761	Ninguno	AdaBoost	64.55%	Combinados	849	Ninguno
RF	79.66%	Combinados	25	RFE	RF	74.03%	Combinados	20	RFE
MLP	78.42%	Matemáticos	761	Ninguno	MLP	71.46%	Matemáticos	25	RFE
SVM	88.06%	Combinados	849	Ninguno	SVM	74.25%	Matemáticos	25	RFE

Los dos modelos con mejor precisión (accuracy) son el modelo SVM que fue entrenado con la totalidad del conjunto de datos matemáticos y obtuvo una precisión del 74.25% y el modelo Random Forest que fue entrenado con un subconjunto de los datos combinados y obtuvo una precisión del 74.03%. El modelo Random Forest fue entrenado con un conjunto de parámetros específicos; esto es, se utilizó un número predeterminado de 100 estimadores (árboles) en el bosque. Además, se configuró el muestreo bootstrap para cada árbol y se fijó la semilla aleatoria (random_state) en 42 para garantizar la reproducibilidad de los resultados. El modelo SVM fue entrenado utilizando el enfoque Uno contra Todos (One-vs-Rest) y se utilizó una búsqueda de hiperparámetros Grid Search, que identificó la combinación óptima para mejorar su rendimiento.

Los hiperparámetros considerados incluyeron el parámetro de regularización (C), que compensa la clasificación correcta de los ejemplos de entrenamiento con la maximización del margen de la función de decisión [117]; el grado del kernel polinómico (degree), indicando el grado del polinomio utilizado en la función kernel polinómica [118]; el coeficiente del kernel (gamma), que define la influencia de un ejemplo en el entrenamiento [117]; y el tipo de kernel

(kernel), especificando la función utilizada para transformar el espacio de características original en uno de mayor dimensión, donde es más fácil separar las clases [118]. Los hiperparámetros óptimos encontrados sugieren que, para este problema específico, un modelo SVM con una fuerte capacidad de regularización ($C=10$), un kernel polinómico de grado 2, un coeficiente de kernel moderado ($\gamma=0.1$), y un kernel radial (rbf) funciona bien en la tarea de clasificación.

En la **Tabla 19**, se muestran los valores de las métricas para cada uno de los subconjuntos (folds) utilizados por el modelo Random Forest en su entrenamiento. Los resultados en la tabla revelan una uniformidad en el desempeño del modelo de clasificación en los 10 conjuntos de datos evaluados. Con una precisión (precision) promedio cercano al 74.31%, el modelo muestra una habilidad sólida para clasificar adecuadamente las instancias. Tanto la precisión como la recuperación (recall) mantienen una consistencia alrededor del 74%, y el puntaje F1 alcanza un promedio de aproximadamente 0.74. Estos valores indican un equilibrio entre la capacidad de prever instancias positivas y evitar falsos positivos. En la **Ilustración 27**, se visualizan los valores de la tabla en gráficos de línea.

Tabla 19

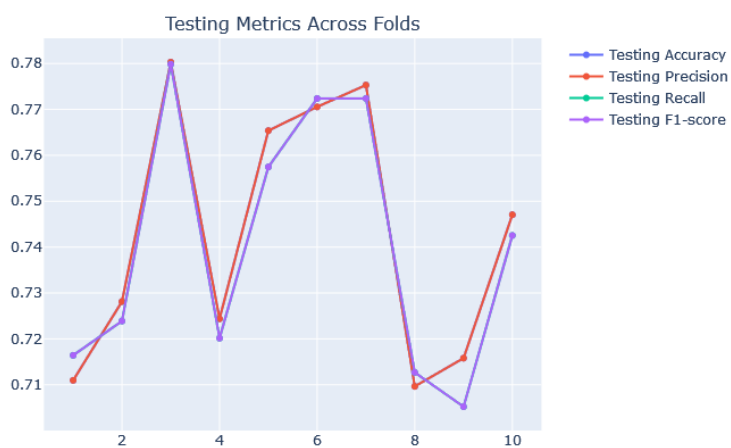
Valores de métricas a través de los subconjuntos (folds)

Fold	Accuracy	Precision	Recall	F1
1	71.64	0.71	0.72	0.71
2	72.39	0.73	0.72	0.72
3	77.99	0.78	0.78	0.78

4	72.01	0.72	0.72	0.72
5	75.75	0.77	0.76	0.75
6	77.24	0.77	0.77	0.77
7	77.24	0.78	0.77	0.77
8	71.27	0.71	0.71	0.7
9	70.52	0.72	0.71	0.7
10	74.25	0.75	0.74	0.74

Ilustración 27

Visualización del comportamiento de métricas a través de los subconjuntos (folds)



En la **Tabla 20**, se muestran los valores de distintas métricas para las tres clases existentes, la clase 0 correspondiente al grupo de sujetos sanos, la clase 1 al grupo de sujetos con EP predominante en la zona izquierda y la clase 2 que corresponde al grupo de sujetos con EP

predominante en la zona derecha. La clase 0 exhibe una mejor combinación de precisión (0.75) y recuperación (0.85), reflejada en una alta puntuación F1 de 0.8. En contraste, la clase 1 muestra una recuperación más baja (0.56), lo que afecta su puntuación F1 (0.64). La clase 2 presenta un equilibrio entre precisión (0.73) y recuperación (0.73), con una puntuación F1 de 0.73. Las métricas macro (0.74 de precisión promedio, 0.71 de recuperación promedio y 0.72 de puntuación F1 promedio) proporcionan una visión general del rendimiento, mientras que las métricas ponderadas (0.74 de precisión ponderada, 0.74 de recuperación ponderada y 0.74 de puntuación F1 ponderada) consideran el impacto relativo de cada clase en función de su soporte.

Tabla 20

Valores de métricas para el modelo SVM

	precision	recall	f1-score	support
0	0.75	0.85	0.8	233
1	0.74	0.56	0.64	123
2	0.73	0.73	0.73	180
Macro Average	0.74	0.71	0.72	536
Weighted Average	0.74	0.74	0.74	536
Overall accuracy	0.74			

La **Ilustración 29** presenta la matriz de confusión del modelo SVM, que se centra en el conjunto de pruebas. Aquí, las clases 0, 1 y 2 del modelo SVM se alinean con las clases 1, 2 y 3 del modelo Random Forest (Ver **Ilustración 28**), respectivamente. Ambos modelos comparten el patrón de mayor acierto para la Clase 0, seguido por la Clase 2 y con menor rendimiento en la Clase 3. La comparación de estas matrices de confusión subraya la consistencia en el rendimiento de ambas arquitecturas de aprendizaje en las clases específicas.

Ilustración 28

Matriz de confusión para el modelo Random Forest

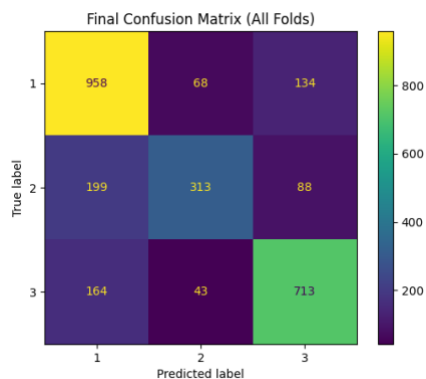
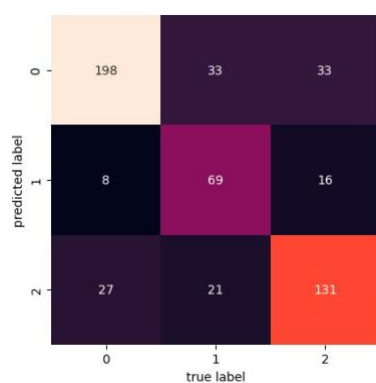


Ilustración 29

Matriz de confusión del modelo SVM elegido



Finalmente, comparando los resultados del modelo SVM y Random Forest, ambos modelos tuvieron un rendimiento bastante similar en términos de exactitud (accuracy), SVM alcanzó un 74.25% y Random Forest obtuvo un 74.03%. La precisión y la puntuación F1 fueron ligeramente más altas en SVM, pero la recuperación fue un poco más baja en comparación con Random Forest. Dada la exactitud superior del modelo SVM, se eligió a este para ser incorporado en la solución final.

3.3 Pruebas realizadas a la página web

La metodología seguida para evaluar la aplicación web se enfocó en medir el tiempo que los usuarios requerían para poder llevar a cabo dos acciones específicas en la aplicación. Estas acciones fueron la carga de un archivo de diagnóstico del paciente para determinar la presencia de Parkinson y la capacidad de agregar una nueva persona al sistema.

La metodología empleada para evaluar la aplicación web se centró en medir el tiempo que los usuarios necesitaban para llevar a cabo dos acciones específicas en la aplicación, además de completar una breve encuesta detallada en el Anexo 1. Estas acciones consistieron en cargar un archivo de diagnóstico del paciente para determinar la presencia de Parkinson y la capacidad de agregar una nueva persona al sistema.

Tabla 21

Información recolectada sobre las pruebas de cargar un diagnóstico del paciente.

Usuarios(n)	Tiempo	Nivel de Dificultad	¿Tuvo problemas?
1	40	2	No
2	54	1	Si
3	46	1	No
4	40	1	No
5	37	1	No
6	48	2	No
7	40	1	Si
8	70	3	Si
9	36	2	No

10	50	2	Si
----	----	---	----

Una vez recopilado el tiempo empleado y las respuestas de la encuesta, se procedió a detallar todos los resultados en la **Tabla 21** correspondiente a las pruebas de carga de un archivo de diagnóstico del paciente.

En la retroalimentación recopilada en la **Tabla 21**, se observa que la mayoría de los usuarios no experimentó dificultades significativas al cargar archivos, con la mayoría calificando su nivel de dificultad como 1. Asimismo, la mayoría de los usuarios no reportó problemas, excepto tres personas que encontraron dificultades al agregar comentarios a los diagnósticos, ya que no quedaba claro si debían subir un solo comentario o varios.

Ilustración 30

Resultado de promedio de segundos para subir un archivo



En la **ilustración 30**, se aprecia que el tiempo promedio necesario para cargar un archivo y recibir el diagnóstico es de aproximadamente 45 segundos. Este resultado demuestra que el

proceso se encuentra dentro de los rangos aceptables, lo que sugiere que llevar a cabo esta acción es relativamente sencillo e intuitivo, sin complicaciones significativas.

La siguiente prueba consistió en agregar un nuevo usuario en el sistema, al igual que en el caso anterior. Se registró el tiempo empleado y se recopilaron las respuestas de la encuesta al **anexo 1** en la **tabla 22**.

Tabla 22

Información recolectada sobre las pruebas realizadas a agregar un usuario al sistema.

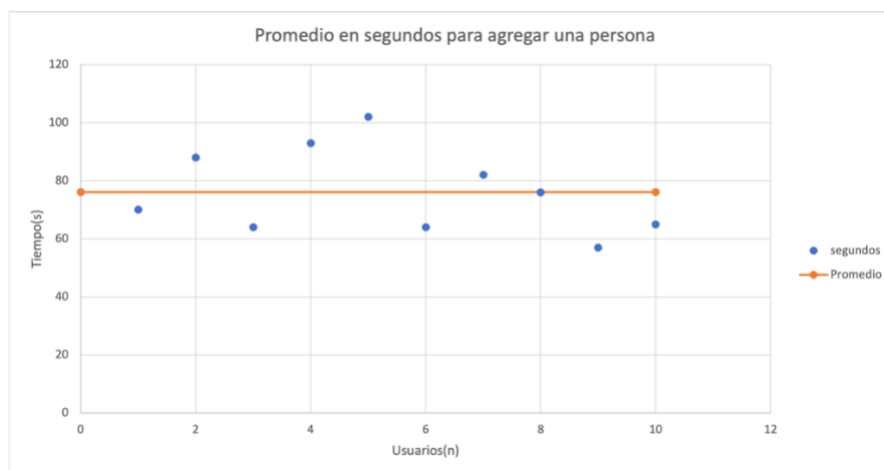
Usuarios(n)	Tiempo	Nivel de Dificultad	¿Tuvo problemas?
1	70	3	No
2	88	1	No
3	64	2	Si
4	93	3	No
5	102	2	No
6	64	1	No
7	82	1	Si
8	76	1	Si
9	57	1	No
10	65	1	No

En la retroalimentación recopilada de la **tabla 22**, se observa que no fue particularmente difícil de realizar, con un promedio de dificultad de 1.6 para realizar esta actividad. En cuanto a los problemas, solo tres personas tuvieron dificultades al intentar encontrar el botón para agregar

usuarios. Esto se debió a que el color del botón era azul, lo que hizo que pareciera que la opción estaba bloqueada, mientras que los demás participantes no encontraron ningún problema.

Ilustración 31

Resultado de promedio en segundos para agregar una persona



En la **Ilustración 31**, se observa que el tiempo promedio requerido para agregar una persona es de 76 segundos. Este proceso, que involucra más pasos en comparación con la acción anterior, se encuentra dentro de los rangos aceptables. Esto se debe a que implica ingresar los datos del usuario y verificar que no existan duplicados, lo cual justifica el tiempo adicional necesario para llevar a cabo esta tarea de manera

CAPÍTULO 4

4. Conclusiones y recomendaciones

En el desarrollo de este proyecto, se empleó un conjunto de datos obtenido de experimentos realizados en sujetos sanos, así como en sujetos que padecían EP predominante en el hemisferio derecho e izquierdo utilizando la tableta. Estos datos fueron utilizados para entrenar y validar los modelos de aprendizaje profundo y poco profundo.

Antes del entrenamiento de los modelos, se realizó un preprocesamiento de los datos de escritura. Para los modelos de aprendizaje poco profundo, se crearon tres conjuntos: uno con características médicas, otro con características matemáticas, y un tercero combinando ambos. Para el conjunto de características matemáticas, se utilizó la biblioteca "tsfresh" [107] para generar una variedad de características de series temporales. Las características médicas fueron generadas a partir de parámetros que incluyeron posición, velocidad, aceleración y jerk, así como métricas de trazo como NCV y NCA. Se calcularon métricas básicas y estadísticas para las dimensiones 'x' y 'y', y se aplicaron siete medidas estadísticas adicionales para enriquecer la representación. Para los modelos de aprendizaje profundo, se crearon representaciones gráficas de las muestras de escritura a partir de sus coordenadas filtradas. Los gráficos se invirtieron y almacenaron como archivos PNG para luego ser comprimidos en formato JPEG. Las imágenes junto a sus etiquetas fueron almacenadas en archivos TFRecord.

Los modelos de aprendizaje poco profundo utilizados para el análisis de atributos fueron SVM, MLP, Random Forest y AdaBoost. Mientras que los modelos de aprendizaje profundo empleados para el análisis de imágenes de diferentes muestras de escritura fueron VGG16, VGG19, ResNet, MobileNet, Inception y Vision Transformers. Todos los modelos mencionados

fueron considerados para ser evaluados para la determinación del modelo a ser incorporado en el producto final.

Se desarrolló una aplicación web donde se integró el modelo final. Esta permite al médico realizar predicciones sobre el estado de un sujeto a partir de datos extraídos de una prueba de escritura. La predicción indica el porcentaje de probabilidad de que el sujeto pertenezca al grupo de sujetos sano, al grupo de sujetos con EP predominante en el lado derecho o al grupo de sujetos con EP predominante en el lado izquierdo.

4.1 Conclusiones

- Los modelos de aprendizaje poco profundo, como SVM y Random Forest, demostraron un mejor rendimiento al capitalizar de manera más efectiva los datos de las muestras de escritura. Lograron los resultados más destacados en términos de precisión tanto para la clasificación binaria como para la clasificación multiclase en comparación con los modelos de aprendizaje profundo.
- La precisión de los modelos de aprendizaje poco profundo se vio notablemente afectada por el método de selección de características Recursive Feature Elimination (RFE). Este método exhibió un desempeño superior en comparación con los otros dos métodos, Mann-Whitney U Test y Kruskal-Wallis, destacándose como el más eficaz en mejorar la calidad del modelo.
- Tras comparaciones, SVM se seleccionó como modelo final por su mayor precisión en la clasificación multiclase. Sin embargo, se destaca el desafío de la memorización en futuras investigaciones subrayando la necesidad de abordar este problema.

4.1 Recomendaciones

Experimentar con diferentes arquitecturas de aprendizaje poco profundo, ya que estas demuestran una capacidad destacada para aprender de manera más efectiva las características de las muestras de escritura. Además, se sugiere, al mismo tiempo, expandir el conjunto de características en busca de características importantes que aporten mejores resultados.

Realizar un análisis de sensibilidad en la elección de parámetros y técnicas de extracción de características. Esto permitiría evaluar cómo distintas configuraciones impactan en el rendimiento de los modelos de aprendizaje poco profundo. Tal análisis puede ser fundamental para ajustar y optimizar los parámetros, buscando un rendimiento aún mejor.

ANEXOS

ANEXO 1

Encuesta Realizada

A continuación, se presenta el cuestionario completo utilizado en la encuesta realizada como parte de esta investigación. El objetivo principal de la encuesta fue recopilar información sobre la experiencia de los usuarios con la aplicación "Neuroscribe". Los participantes fueron divididos en dos grupos de 10 personas cada uno. El primer grupo asumió el rol de "doctor" al utilizar la aplicación, mientras que el segundo grupo asumió el rol de "administrador". Los participantes fueron invitados a calificar la dificultad de realizar una actividad utilizando la aplicación, informar sobre cualquier problema que encontraron durante su uso y compartir sus opiniones sobre lo que más les gustó de la misma.

Instrucciones: Por favor, responda las siguientes preguntas con honestidad y en función de su experiencia al utilizar la aplicación Neuroscribe.

Pregunta 1: Califique la dificultad que tuvo realizando su actividad utilizando la aplicación (del 1 al 5)

Por favor, seleccione un número del 1 al 5, donde 1 representa "Muy fácil" y 5 representa "Muy difícil".

1 (Muy fácil)

2 (Fácil)

3 (Moderado)

4 (Difícil)

5 (Muy difícil)

Pregunta 2: ¿Encontró algún problema haciendo su actividad utilizando la aplicación?
Menciónelos.

Por favor, describa cualquier problema que haya experimentado al utilizar la aplicación para llevar a cabo su actividad. Puede mencionar aspectos técnicos, de usabilidad, o cualquier otra dificultad que haya encontrado.

No encontré ningún problema.

Sí, encontré problemas. (Por favor, describa los problemas a continuación):

[Texto abierto para que el encuestado escriba sus respuestas]

Pregunta 3: ¿Qué es lo que más le gustó de la aplicación?

Por favor, comparta con nosotros lo que más le gustó de la aplicación Neuroscribe. Sus comentarios nos ayudarán a enfocarnos en los aspectos positivos y a seguir mejorando.

[Texto abierto para que el encuestado escriba sus respuestas]

BIBLIOGRAFÍA

- [1] de Lau LM, Breteler MM. Epidemiology of Parkinson's disease. *Lancet Neurol* 2006;5(6):525–35.
- [2] von Campenhausen S, Bornschein B, Wick R, Botzel K, Sampaio C, Poewe W, et al. Prevalence and incidence of Parkinson's disease in Europe. *Eur Neuropsychopharm* 2005;15(4):473–90.
- [3] Contreras-Vidal JL, Stelmach GE. Effects of parkinsonism on motor control. *Life Sci* 1995;58(3):165–76.
- [4] Hughes AJ, Daniel SE, Ben-Shlomo Y, Lees AJ. The accuracy of diagnosis of parkinsonian syndromes in a specialist movement disorder service. *Brain* 2002;125(4):861–70, <http://dx.doi.org/10.1093/brain/awf080>.
- [5] P. Pastor y E. Tolosa, "La enfermedad de Parkinson: diagnóstico y avances en el conocimiento de la etiología y en el tratamiento," en *Med. Integral*, vol. 37, no. 3, pp. 104-117, 2001.
- [6] Fuentes, C.T, Mostofsky, S.H., Bastian, A.J., 2009. Children with autism show specific handwriting impairments. *Neurology* 73, 1532–1537.
- [7] Kushki, A., Chau, T., Anagnostou, E., 2011. Handwriting difficulties in children with autism spectrum disorders: A scoping review. *Journal of autism and developmental disorders* 41, 1706–1716.
- [8] Saunders-Pullman R, Derby C, Stanley K, Floyd A, Bressman S, Lipton RB, et al. Validity of spiral analysis in early Parkinson's disease. *Mov Disord* 2008;23(4):531–7.
- [9] Teulings, H., Contreras-Vidal, J.L., Stelmach, G., Adler, C.H., 2002. Adaptation of handwriting size under distorted visual feedback in patients with

parkinson's disease and elderly and young controls. *Journal of Neurology, Neurosurgery & Psychiatry* 72, 315–324.

[10] Bidet-Ildei C, Pollak P, Kandel S, Fraix V, Orliaguet J-P. Handwriting in patients with Parkinson's disease: effect of L-DOPA and stimulation of the sub-thalamic nucleus on motor anticipation. *Hum Mov Sci* 2011;30(4):783–91.

[11] Slavin, M.J., Phillips, J.G., Bradshaw, J.L., Hall, K.A., Presnell, I., 1999. Consistency of handwriting movements in dementia of the alzheimer's type: a comparison with huntington's and parkinson's diseases. *Journal of the International Neuropsychological Society* 5, 20–25.

[12] Drotar P, Mekyska J, Rektorova I, Masarova L, Smekal Z, Faundez-Zanuy M. A new modality for quantitative evaluation of Parkinson's disease: in-air movement. In: 2013 IEEE 13th international conference on Bioinformatics and bioengineering (BIBE). 2013. p. 1–4.

[13] Rosenblum S, Samuel M, Zlotnik S, Erih I, Schlesinger I. Handwriting as an objective tool for parkinsons disease diagnosis. *J Neurol* 2013;260(9):2357–61.

[14] UnluA, Brause R, Krakow K. Handwriting analysis for diagnosis and prognosis of parkinsons disease. In: Maglaveras N, Chouvarda I, Koutkias V, Brause R, editors. *Biological and medical data analysis*, Vol. 4345 of lecture notes in computer science. Berlin, Heidelberg: Springer; 2006. p. 441–50

[15] Kraus PH, Hoffmann A. Spiralometry: computerized assessment of tremor amplitude based on spiral drawing. *Mov Disord* 2010;25(13):2164–70, <http://dx.doi.org/10.1002/mds.23193>. [16]

Stanley K, Hagenah J, Bruggemann N, Reetz K, Severt L, Klein C, et al. Digitized spiral analysis is a promising early motor marker for Parkinson's disease. *Parkinsonism Relat Disord* 2010;16(3):233–4

- [16] Stanley K, Hagenah J, Bruggemann N, Reetz K, Severt L, Klein C, et al. Digitized spiral analysis is a promising early motor marker for Parkinson's disease. *Parkinsonism Relat Disord* 2010;16(3):233–4
- [17] Smits EJ, Tolonen AJ, Cluitmans L, van Gils M, Conway BA, Zietsma RC, et al. Standardized handwriting to assess bradykinesia, micrographia and tremor in parkinson's disease. *PLOS ONE* 2014; 9:1–8
- [18] Teulings H-L, Contreras-Vidal JL, Stelmach GE, Adler CH. Parkinsonism reduces coordination of fingers, wrist, and arm in fine motor control. *Exp Neurol* 1997;146(1):159–70.
- [19] Sesa-Nogueras E, Faundez-Zanuy M, Mekyska J. An information analysis of in-air and on-surface trajectories in online handwriting. *Cogn Comput* 2012;4(2):195–205, <http://dx.doi.org/10.1007/s12559-011-9119-y>.
- [20] Schroter, A., Mergl, R., B ¨urger, K., Hampel, H., M ¨oller, H.J., Hegerl, U., 2003. Kinematic analysis of handwriting movements in patients with alzheimers disease, mild cognitive impairment, depression, and healthy subjects. *Dementia and Geriatric Cognitive Disorders* 15, 132–142.
- [21] Caligiuri, M.P., Mohammed, L.A., 2012. *The neuroscience of handwriting: Applications for forensic document examination*. CRC Press.
- [22] Waton, J., 1997. Handwriting changes due to aging and parkinson's syndrome. *Forensic science international* 88, 197–214.
- [23] Rosenblum, S., Engel-Yeger, B., Fogel, Y., 2013a. Age-related changes in executive control and their relationships with activity performance in handwriting. *Human movement science* 32, 363–376.

- [25] Derkinderen, P., Dupont, S., Vidal, J.S., Chedru, F., Vidailhet, M., 2002. Micrographia secondary to lenticular lesions. *Movement disorders* 17, 835–837.
- [26] Berardelli, A., Rothwell, J.C., Thompson, P.D., Hallett, M., 2001. Pathophysiology of bradykinesia in parkinson's disease. *Brain* 124, 2131–2146.
- [27] Zhi, N., Jaeger, B.K., Gouldstone, A., Frank, S., Sipahi, R., 2015. Objective quantitative assessment of movement disorders through analysis of static handwritten characters, in: *ASME 2015 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers. pp. V001T16A006–V001T16A006.
- [28] Palmerini, L., Rocchi, L., Mellone, S., Valzania, F., Chiari, L., 2011. Feature selection for accelerometer-based posture analysis in parkinson's disease. *IEEE Transactions on Information Technology in Biomedicine* 15, 481–490.
- [29] Ünlü, A., Brause, R., Krakow, K., 2006. Handwriting analysis for diagnosis and prognosis of parkinsons disease, in: *International Symposium on Biological and Medical Data Analysis*, Springer. pp. 441–450.
- [30] Drotar, P., Mekyska, J., Rektorova, I., Masarova, L., Smékal, Z., Faundez-Zanuy, M., 2013a. A new modality for quantitative evaluation of parkinson's disease: In-air movement, in: *IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2013, IEEE. pp. 1–4.
- [31] Drotar, P., Mekyska, J., Rektorova, I., Masarova, L., Smékal, Z., Faundez-Zanuy, M., 2014. Analysis of in-air movement in handwriting: A novel marker for parkinson's disease. *Computer methods and programs in biomedicine* 117, 405–411.
- [32] Rosenblum, S., Samuel, M., Zlotnik, S., Erikh, I., Schlesinger, I., 2013b. Handwriting as an objective tool for parkinsons disease diagnosis. *Journal of neurology* 260, 2357-2361.

- [33] Graça, R., e Castro, R.S., Cevada, J., 2014. Parkdetect: Early diagnosing parkinson's disease, in: IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2014, IEEE. pp. 1–6.
- [34] Pereira, C.R., Pereira, D.R., da Silva, F.A., Hook, C., Weber, S.A., Pereira, L.A., Papa, J.P., 2015. A step towards the automated diagnosis of parkinson's disease: Analyzing handwriting movements, in: IEEE 28th International Symposium on Computer-Based Medical Systems (CBMS), 2015, IEEE. pp. 171–176.
- [35] Pereira, C.R., Weber, S.A., Hook, C., Rosa, G.H., Papa, J.P., 2016. Deep learning-aided parkinson's disease diagnosis from handwritten dynamics, in: 29th SIBGRAPI Conference on Graphics, Patterns, and Images (SIBGRAPI), 2016, IEEE. pp. 340–346
- [36] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1989. Handwritten digit recognition with a backpropagation network, in: Neural Information Processing Systems (NIPS).
- [37] LeCun, Y., Bottou, L., Bengio, Y., Haner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- [38] Moetesum, M., Siddiqi, I., Masroor, U., Djeddi, C., 2015. Automated scoring of bender gestalt test using image analysis techniques, in: 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, IEEE. pp. 666–670.
- [39] Drotar, P., Mekyska, J., Smékal, Z., Rektorova, I., Masarova, L., Faundez-Zanuy, M., 2013b. Prediction potential of different handwriting tasks for diagnosis of parkinson's, in: E-Health and Bioengineering Conference (EHB), 2013, IEEE. pp. 1–4

- [40] Página web de la Fundación de Parkinson. 'Enfermedad de Parkinson.' Disponible en: <https://w600ww.parkinson.org/library/fact-sheets/enfermedad-parkinson>
- [41] Alalayah, K. M., Senan, E. M., Atlam, H. F., Ahmed, I. A., & Shatnawi, H. S. A. (2023). Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method. *Diagnostics*, 13(11), 1924.
- [42] Drotár, P., Mekyska, J., Rektorová, I., Masarová, L., Smékal, Z., & Faundez-Zanuy, M. (2016). Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease. *Artificial intelligence in Medicine*, 67, 39-46.
- [43] Cho, C.-W., et al. (2009). "A vision-based analysis system for gait recognition in patients with Parkinson's disease." *Exp4ert Systems with applications* 36(3): 7033-7039.
- [44] Chakraborty, S., Aich, S., Sim, J. S., Han, E., Park, J., & Kim, H.-C. (2020). Parkinson's Disease Detection from Spiral and Wave Drawings using Convolutional Neural Networks: A Multistage Classifier Approach. In *22nd International Conference on Advanced Communication Technology (ICACT)* (pp. 298-303). Phoenix Park, Korea (South). DOI: 10.23919/ICACT48636.2020.9061497.
- [45] Lola Díaz-Feliz. (2022) "Valoración computarizada de la escritura en la enfermedad de Parkinson".
- [46] Ruiz Barquín, O. (2023). El papel de la enfermera en el proceso de valoración y abordaje de un paciente con enfermedad de Parkinson.
- [47] Mantri, S., & Morley, J. F. (2018). Prodromal and early Parkinson's disease diagnosis. *Practical Neurology*, 35, 28-31.

[48] Postuma, R. B., & Berg, D. (2016). Advances in markers of prodromal Parkinson disease. **Nature Reviews Neurology**, 12(11), 622–634.

[49] Getting diagnosed. Parkinson's Foundation. (n.d.).

<https://www.parkinson.org/understanding-parkinsons/getting-diagnosed>

[50] Carvalho, V., Cunha, C.V., Massano, J. (2018). Parkinson's Disease: Contemporary Concepts and Clinical Management. In: Galimberti, D., Scarpini, E. (eds) *Neurodegenerative Diseases*. Springer, Cham. https://doi.org/10.1007/978-3-319-72938-1_17

[51] Zham, P., Kumar, D. K., Dabnichki, P., Raghav, S., & Keloth, S. M. (2016). Dynamic handwriting analysis for assessing movement disorder. In 13th International Conference of Applied Computing.

[52] Beach, T. G., & Adler, C. H. (2018). Importance of low diagnostic Accuracy for early Parkinson's disease. *Movement disorders: official journal of the Movement Disorder Society*, 33(10), 1551–1554. <https://doi.org/10.1002/mds.27485>

[53] Inzelberg, R., Plotnik, M., Kadmon Harpaz, N., & Flash, T. (2016). Micrographia, much beyond the writer's hand. *Parkinsonism & Related Disorders*, 26, 1-9. <https://doi.org/10.1016/j.parkreldis.2016.03.003>.

[54] Thomas, M., Lenka, A., & Kumar Pal, P. (2017). Handwriting Analysis in Parkinson's Disease: Current Status and Future Directions. *Movement disorders clinical practice*, 4(6), 806–818. <https://doi.org/10.1002/mdc3.12552>

[55] Drotár, P., Mekyska, J., Rektorová, I., Masarová, L., Smékal, Z., & Faundez-Zanuy, M. (2015). Decision support framework for Parkinson's disease based on novel handwriting markers. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, 23(3), 508-516. <https://doi.org/10.1109/TNSRE.2014.2359997>

[56] Taleb, C., Khachab, M., Mokbel, C., & Likforman-Sulem, L. (2017). Feature selection for an improved Parkinson's disease identification based on handwriting. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)* (pp. 52-56). Nancy, France. doi: 10.1109/ASAR.2017.8067759.

[57] Mucha, J., Mekyska, J., Galaz, Z., Faundez-Zanuy, M., Lopez-de-Ipina, K., Zvoncak, V., Kiska, T., Smekal, Z., Brabenec, L., & Rektorova, I. (2018). Identification and Monitoring of Parkinson's Disease Dysgraphia Based on Fractional-Order Derivatives of Online Handwriting. *Applied Sciences*, 8, 2566. <https://doi.org/10.3390/app8122566>

[58] Tolosa, E., Garrido, A., Scholz, S. W., & Poewe, W. (2021). Challenges in the diagnosis of Parkinson's disease. *The Lancet. Neurology*, 20(5), 385–397. [https://doi.org/10.1016/S1474-4422\(21\)00030-2](https://doi.org/10.1016/S1474-4422(21)00030-2)

[59] Thomas, M., Lenka, A., & Kumar Pal, P. (2017). Handwriting Analysis in Parkinson's Disease: Current Status and Future Directions. *Movement disorders clinical practice*, 4(6), 806–818. <https://doi.org/10.1002/mdc3.12552>

[60] Öksüz, N., Öztürk, Ş., & Doğu, O. (2022). Future Prospects in Parkinson's Disease Diagnosis and Treatment. *Noro psikiyatri arsivi*, 59(Suppl 1), S36–S41. <https://doi.org/10.29399/npa.28169>

[61] Landers, M., Saria, S., & Espay, A. J. (2021). Will Artificial Intelligence Replace the Movement Disorders Specialist for Diagnosing and Managing Parkinson's Disease? *Journal of Parkinson's disease*, 11(s1), S117–S122. <https://doi.org/10.3233/JPD-212545>

[62] Saavedra Moreno, Juan Sebastián, Millán, Paula Andrea, & Buriticá Henao, Omar Fredy. (2019). Introducción, epidemiología y diagnóstico de la enfermedad de Parkinson. *Acta Neurológica Colombiana*, 35(Suppl. 1), 2-10. <https://doi.org/10.22379/24224022244>

[63] Pereira, C. R., Weber, S. A. T., Hook, C., Rosa, G. H., & Papa, J. P. (2016). Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics. In 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 340-346). Sao Paulo, Brazil. doi: 10.1109/SIBGRAPI.2016.054.

[64] Taleb, C., Khachab, M., Mokbel, C., & Likforman-Sulem, L. (2017, April). Feature selection for an improved Parkinson's disease identification based on handwriting. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR) (pp. 52-56). IEEE.

[65] Instituto de Investigación Sanitaria Fundación Jiménez Díaz (2022). Valoración computarizada de la escritura en la enfermedad de Parkinson [PDF]. Fundación Jiménez Díaz. [https://www.fjd.es/iis-fjd/es/areas-grupos-](https://www.fjd.es/iis-fjd/es/areas-grupos-investigacion/neurociencias/neurologia/presentaciones-neurologia-2022.ficheros/2925597-221222_04%20Escritura%20en%20EP_Sesión%20del%20Area%20de%20Neurociencias.pdf)

[investigacion/neurociencias/neurologia/presentaciones-neurologia-2022.ficheros/2925597-221222_04%20Escritura%20en%20EP_Sesión%20del%20Area%20de%20Neurociencias.pdf](https://www.fjd.es/iis-fjd/es/areas-grupos-investigacion/neurociencias/neurologia/presentaciones-neurologia-2022.ficheros/2925597-221222_04%20Escritura%20en%20EP_Sesión%20del%20Area%20de%20Neurociencias.pdf)

[66] Pisner, D. A., Schnyer, D. M. (2020). Support vector machine. En A. Mechelli & S. Vieira (Eds.), Machine Learning (pp. 101-121). Academic Press. <https://doi.org/10.1016/B978-0-12-815739-8.00006-7>

[67] Williams, G. (2011). Support Vector Machines. In Data Mining with Rattle and R. Springer. https://doi.org/10.1007/978-1-4419-9890-3_14

[68] Pradhan, A. (2012). Support vector machine-a survey. International Journal of Emerging Technology and Advanced Engineering, 2(8), 82-85.

[69] Kégl, B. (2009). Introduction to adaboost.

[70] Hotho, A., Nürnberger, A., & Paab, G. (2005). A brief survey of text mining. LDV Forum-GLDV Journal of Computational Linguistics and Language Technology, 20(1), 19-62.

[71] Nazarenko, E., Varkentin, V., & Polyakova, T. (2019). Features of Application of Machine

Learning Methods for Classification of Network Traffic (Features, Advantages, Disadvantages).

In 2019 International Multi-Conference on Industrial Engineering and Modern Technologies

(FarEastCon) (pp. 1-5). Vladivostok, Russia. <https://doi.org/10.1109/FarEastCon.2019.8934236>

[72] Louppe, G. (2014). Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502.

[73] Kong, Y., & Yu, T. (2018). A Deep Neural Network Model using Random Forest to Extract Feature Representation for Gene Expression Data Classification. *Scientific reports*, 8(1), 16477. <https://doi.org/10.1038/s41598-018-34833-6>

[74] Ramezani, R., Peymanfar, A., & Ebrahimi, S. B. (2019). An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: An application in Tehran stock exchange market. *Applied Soft Computing*, 82, 105551. <https://doi.org/10.1016/j.asoc.2019.105551>

[75] Abirami, S., & Chitra, P. (2020). Energy-efficient edge-based real-time healthcare support system. In P. Raj & P. Evangeline (Eds.), **Advances in Computers** (Vol. 117, Issue 1, pp. 339-368). Elsevier. <https://doi.org/10.1016/bs.adcom.2019.09.007>

[76] Edgar, T. W., & Manz, D. O. (2017). Exploratory Study. In T. W. Edgar & D. O. Manz (Eds.), **Research Methods for Cyber Security** (pp. 95-130). Syngress. <https://doi.org/10.1016/B978-0-12-805349-2.00004-2>

[77] Subasi, A. (2020). Machine learning techniques. In A. Subasi (Ed.), **Practical Machine Learning for Data Analysis Using Python** (pp. 91-202). Academic Press. <https://doi.org/10.1016/B978-0-12-821379-7.00003-5>

[78] Yukio. (2018, agosto 29). Regressão Logística: Conceitos Essenciais e Modelo.

ESTATSITE. <https://estatsite.com.br/2018/08/29/regressao-logistica-conceitos-e-formula/>

- [79] Kisner, H., Ding, Y., Thomas, U., & Li, Q. (Eds.). (2022). Capacitive material detection with machine learning for robotic grasping applications. En Q. Li, S. Luo, Z. Chen, C. Yang, & J. Zhang (Eds.), *Tactile Sensing, Skill Learning, and Robotic Dexterous Manipulation* (pp. 59-79). Academic Press. <https://doi.org/10.1016/B978-0-32-390445-2.00011-8>.
- [80] Bezdan, T., & Bačanin Džakula, N. (2019). Convolutional Neural Network Layers and Architectures. En *International Scientific Conference on Information Technology and Data Related Research* (pp. 445–451). <https://doi.org/10.15308/Sinteza-2019-445-451>.
- [81] Singh, S. A., Meitei, T. G., & Majumder, S. (2020). Short PCG classification based on deep learning. En *Deep Learning Techniques for Biomedical and Health Informatics* (pp. 141–164). Elsevier Inc. <https://doi.org/10.1016/B978-0-12-819061-6.00006-9>
- [82] Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Computer vision for human-machine interaction. En *Computer Vision for Assistive Healthcare* (pp. 127–145). <https://doi.org/10.1016/B978-0-12-813445-0.00005-8>.
- [83] Bhatt, D., et al. (2021). CNN variants for computer vision: History, architecture, application, challenges, and future scope. *Electron.*, 10(20), 2470. <https://doi.org/10.3390/s19010217>.
- [84] Wang, Z. J., et al. (2021). CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 1396–1406. https://doi.org/10.1162/neco_a_00990.
- [85] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556. <https://arxiv.org/abs/1409.1556>.
- [86] Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. En *2021 International Conference*

on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India (pp. 96-99). <https://doi.org/10.1109/CENTCON52345.2021.9687944>.

[87] Chatterjee, C. C. (2019, 31 de julio). Fundamentos de la CNN clásica: Cómo funciona una CNN clásica (Red Neuronal Convolutacional). Towards Data Science. Recuperado de <https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add>

[88] Xu, Y., Zhou, Y., Sekula, P., & Ding, L. (2021). Machine learning in construction: From shallow to deep learning. *Developments in the Built Environment*, 6, 100045. <https://doi.org/10.1016/j.dibe.2021.100045>

[89] Zhong, G., Ling, X., & Wang, L. N. (2019). From shallow feature learning to deep learning: Benefits from the width and depth of deep architectures. *WIREs Data Mining and Knowledge Discovery*, 9(1), e1255. <https://doi.org/10.1002/widm.1255>

[90] Pasupa, K., & Sunhem, W. (2016). A comparison between shallow and deep architecture classifiers on small dataset. In 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 1-6). Yogyakarta, Indonesia. doi: 10.1109/ICITEED.2016.7863293.

[91] Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & Rajabi, M. J. (2014). Advantage and drawback of support vector machine functionality. In 2014 International Conference on Computer, Communications, and Control Technology (I4CT) (pp. 63-65). Langkawi, Malaysia. DOI: 10.1109/I4CT.2014.6914146.

[92] Duan, B. (2019). Research on Application of Support Vector Machine in Machine Learning. *Journal of Electronic Research and Application*, 3. <https://doi.org/10.26689/jera.v3i4.916>.

[93] Moguerza, J. M., & Muñoz, A. (2006). Support Vector Machines with Applications. *Statistical Science*, 21(3), 322-336. <https://doi.org/10.1214/088342306000000493>

- [94] Sinha, D., & El-Sharkawy, M. (2019, October). Thin mobilenet: An enhanced mobilenet architecture. In 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON) (pp. 0280-0285). IEEE
- [95] Google Cloud. (s. f.). Inception V3: Advanced Guide. Recuperado de <https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=es-419>
- [96] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
- [97] Pintos, N. U., Lacomí, H., & Lavorato, M. (2022). B-VGG16: Red Neuronal de Convolución cuantizada binariamente para la clasificación de imágenes. *Elektron*, 6(2), 107-114.
- [98] Díaz-Gaxiola, E., Morales-Casas, Z. E., Castro-López, O., Beltrán-Gutiérrez, G., López, I. F. V., & Rendón, A. Y. (2019). Estudio comparativo de arquitecturas de CNNs en hojas de Pimiento Morrón infectadas con virus PHYVV o PEPGMV. *Res. Comput. Sci.*, 148(7), 289-303.
- [99] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s), 1-41.
- [100] Han, B., Hu, M., Wang, X., & Ren, F. (2022). A triple-structure network model based upon MobileNet V1 and multi-loss function for facial expression recognition. *Symmetry*, 14(10), 2055.
- [101] Great Learning. (2022, January 5). Everything you need to know about VGG16 - Great Learning - Medium. Medium. <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>
- [102] Clifford K. Yang(2018,January 5). https://www.researchgate.net/figure/illustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356

- [103] Modi, R. (2022, February 25). ResNet — Understand and Implement from scratch - Analytics Vidhya - Medium. Medium. <https://medium.com/analytics-vidhya/resnet-understand-and-implement-from-scratch-d0eb9725e0db>
- [104] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2010). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv 2020. arXiv preprint arXiv:2010.11929.
- [105] What is Random Forest? | IBM. (n.d.). <https://www.ibm.com/topics/random-forest>
- [106] Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). **Neurocomputing, 307**, 72-77. doi: 10.1016/j.neucom.2018.03.067.
- [107] Chen, X.-W., & Jeong, J. C. (2007). Enhanced recursive feature elimination. In **Sixth International Conference on Machine Learning and Applications (ICMLA 2007) ** (pp. 429-435). Cincinnati, OH, USA. doi: 10.1109/ICMLA.2007.35.
- [108] Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. **Annals of Mathematical Statistics, 18**, 50–60.
- [109] Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association, 47**, 583–621.
- [110] Salcedo-Sanz, S., Pérez-Aracil, J., Ascenso, G., Del Ser, J., Casillas-Perez, D., Kadow, C., ... Castelletti, A. (2022). Analysis, Characterization, Prediction and Attribution of Extreme Atmospheric Events with Machine Learning: a Review. arXiv. <https://doi.org/10.48550/arXiv.2207.07580>
- [111] Google colab. (s/f). Google.com. Recuperado el 30 de enero de 2024, de <https://research.google.com/colaboratory/faq.html>

- [112] Anaconda Software Distribution. (2020). Anaconda Documentation. Anaconda Inc. Retrieved from <https://docs.anaconda.com/>.
- [113] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
- [114] Abadi, M., et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org. <https://www.tensorflow.org/>
- [115] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. <https://www.citeulike.org/article/2236121>
- [116] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- [117] 1.4. Support vector machines. (s/f). Scikit-Learn. Recuperado el 30 de enero de 2024, de <https://scikit-learn.org/stable/modules/svm.html>
- [118] Sklearn.Svm.SVC. (s/f). Scikit-Learn. Recuperado el 30 de enero de 2024, de <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- [119] Yangqing, J. (s/f) Caffe. Recuperado el 30 de enero de 2024, de <https://caffe.berkeleyvision.org/>.