



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Modelos de aprendizaje
automático para la detección
temprana de retinopatía
diabética**

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Angel Damian Monroy Mendoza

DIRECTOR DE TESIS

Dr. Ismael Everardo Bárcenas Patiño



Ciudad Universitaria, Cd. Mx., 2026



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado MODELOS DE APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN TEMPRANA DE RETINOPATÍA DIABÉTICA que presenté para obtener el título de INGENIERO MECATRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

ANGEL DAMIAN MONROY MENDOZA
Número de cuenta: 316040707

Agradecimientos

Como individuo que ejerce y aplica los conocimientos adquiridos sobre Inteligencia Artificial, Ciencia de Datos, Aprendizaje Automatizado, etc. No me puedo permitir obviar una regla fundamental dentro de esta material. La protección de datos sensibles es un pilar dentro del ejercicio de aquellos que hacen uso y manipulación de la información de personas. En este sentido y, con excepción de la próxima persona, en este apartado colocaré las iniciales, sumándole la primera letra, de los nombres que fungieron como un apoyo fundamental, en lo personal y académico, que desembocó en la culminación de este primer capítulo de mi trayectoria profesional.

En este sentido, quiero dejar mi eterna gratitud en este trabajo al Dr. Everardo Bárcenas, por todo el apoyo, comprensión, guía y, sobretodo, paciencia, que me brindó sin miramiento durante el lapso que ocupé para realizar mi tesis. Además de ello, me ha brindado valiosas oportunidades y conocimientos que, de no ser por ellos, no me hubiera sido posible llegar a donde estoy ahorita con la misma facilidad con la que sucedieron los acontecimientos en mi vida profesional.

Aprovecho para agradecerles a los integrantes de mi sínodo: Mtro. Pe, Mtro. Ba, Dr. Pe y Dr. Ar, quienes, sin su apoyo, revisión y comentarios en este proceso, de igual forma no me hubiera sido posible concluir este proceso.

Siguiendo con lo descrito en el primer párrafo y tomando un orden cronológico comenzando por mi familia, agradezco profundamente a: Ju, Ma, RoEl, Yo, Jo, Er, Arm, JuGe, Gr, No, Ik, Ci, JuMaIg, Am, Le, Je, He, Ma, Tu, Co, Ch. Siguiendo con personas que conocí y se han convertido en familia para mi, comenzando con primaria: Ed, Hu, Er, Da, Mo, Gu, Ja, Me, PaEd, MaHu. Siguiendo con preparatoria: Ge, Em, Da, Ja, Sa, Se, AnSa, AlPe, Fr. Continuando con Universidad: Ma, YaCo, SoHe, He. Y terminando con personas entre medias que, aunque la vida decidió que tomáramos rumbos diferentes, han significado mucho en mi permanencia y crecimiento personal o académico.

Por último, como cimienta a cerrar ciertos ciclos, quiero dejar constancia de un agradecimiento personal que, para aquellos que han convivido conmigo, sabrán que ha sido un camino muy duro, aspero y complejo para llegar hasta este punto, no solo de este trabajo, sino también de mi paso por este mundo.

Con el corazón.

*Angel Damian Monroy Mendoza
EDOMEX, cd. Ecatepec, Enero del 2026.*

Índice general

Índice general	IV
1. Introducción	1
1.1. Definición del Problema	2
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Particulares	4
1.3. Trabajos Relacionados	5
2. Marco Teórico	9
2.1. Escalogramas	9
2.2. Inteligencia Artificial	10
2.3. Aprendizaje Automatizado	11
2.3.1. Árboles de Decisión y Bosques Aleatorios	12
2.3.2. Regresión Logística	16
2.3.3. Bayes Ingenuo Gaussiano	17
2.3.4. Máquinas de Soporte Vectorial	19
2.4. Aprendizaje Profundo	20
2.4.1. Redes Neuronales	20
2.4.2. Redes Neuronales de Perceptrón Multicapa	21

2.4.3. Redes Neuronales Recurrentes	23
2.4.4. Redes Neuronales Convolucionales	24
2.4.5. Redes Neuronales Generativas Adversarias	25
2.5. Extracción de Características	27
2.6. Métricas de Evaluación	30
3. Experimentación	33
3.1. Preprocesamiento	34
3.2. Extracción de Características	36
3.3. Implementación de Modelos y Optimización de Hiperparámetros	37
3.4. Almacenamiento de Resultados	38
4. Resultados	39
4.1. Descripción	39
4.1.1. Escalogramas	40
4.1.2. Datos Sintéticos	41
4.2. Análisis	42
5. Conclusiones	47

Índice de Algoritmos

1.	Construcción de un Árbol de Decisión	16
----	--	----

Introducción

La *Inteligencia Artificial* (IA) emerge como un concepto dominante en la innovación y la tecnología, ganando terreno continuamente. Es acuñado por primera vez por John McCarthy (1927 - 2011) durante la conferencia *Dartmouth Summer Research Project on Artificial Intelligence* en 1956, quien la definió como [...] *la ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas de computación inteligentes, que comprendan la inteligencia humana*[1]. Esto le valió el título de “Padre de la Inteligencia Artificial”.

Con el tiempo, como es natural en cualquier disciplina emergente, la IA se fragmenta en áreas específicas que se centran en tareas y objetivos particulares. Uno de estos subcampos es el Aprendizaje Automatizado (AA), un término acuñado por *Arthur Samuel* en 1959. Según [2], el Aprendizaje Automatizado implica [...] *programar computadoras para optimizar un criterio de rendimiento utilizando datos de ejemplo o experiencias pasadas*. En resumen, se trata de una rama de la IA que capacita a las computadoras para aprender y hacer predicciones

El Aprendizaje Automatizado tiene diversas aplicaciones, desde finanzas hasta psicología, pasando por la rama del derecho, el entorno social, empresarial y el industrial. [2] resalta su impacto en el ámbito médico, donde los algoritmos de aprendizaje apoyan diagnósticos precisos. Aun dentro de esta esfera, el Aprendizaje Automatizado es versátil, permitiendo desde la identificación del tipo de tumor cutáneo en un nuevo paciente hasta la estimación de la probabilidad de padecer ciertas enfermedades.

Este trabajo tiene su origen a raíz de un proyecto colaborativo entre el Instituto de Neurobiología en el campus de Juriquilla y la Facultad de Ingeniería en Ciudad Universitaria, ambas pertenecientes a la UNAM [3]. Dicho trabajo tiene como objetivo principal el desarrollar varios modelos de Aprendizaje Automatizado mediante diversos enfoques para lograr identificar principios de retinopatía diabética de una forma eficiente y confiable. Además, esta investigación incluye desde el preprocesamiento de datos crudos hasta el diseño, optimización de modelos y análisis de métricas de clasificación para evaluar su rendimiento. En línea con lo comentado anteriormente, esta tesis ofrece una perspectiva innovadora en el diagnóstico temprano de la retinopatía diabética, destacando la simplicidad de detección mediante procedimientos no invasivos y rutinarios, así como técnicas vanguardistas en el campo de la inteligencia artificial.

Sección 1.1.

Definición del Problema

La diabetes tipo 2, según [4], representa más del 90 % de todos los casos de diabetes en el mundo, y en el artículo [5] se describe como una enfermedad crónica que afecta la producción y utilización de insulina en el organismo, lo que resulta en niveles elevados de glucosa en la sangre y alteraciones metabólicas, característicamente en la diabetes tipo 2.

El surgimiento de esta enfermedad se ha hecho una realidad cada vez más latente en la población mexicana, pues según lo señala un reporte de la *International Diabetes Federation* [4], en el año 2021 México alcanzó cifras de 14.1 millones de personas con diabetes, y se proyecta que esta cifra alcance los 21.2 millones para 2045. Estos números representan una alerta roja tanto para el área de la salud como para la economía mexicana, ya que el tratamiento y las consecuencias asociadas a la diabetes tipo 2 representan una carga financiera considerable, lo que desemboca en un problema tanto para la sostenibilidad de la atención médica como para la estabilidad financiera de los pacientes.

Una de las complicaciones graves de la diabetes tipo 2 es la *Retinopatía Diabética* (RD), definida por [6] como una condición en la que los vasos sanguíneos internos de la retina se hinchan, se desprenden y dañan los tejidos nerviosos, causando hemorragias y pérdida de visión. Esta enfermedad puede manifestarse con visión borrosa, áreas oscuras en el campo visual y dificultades para percibir colores. Aunque es posible tratarla en etapas tempranas mediante cirugía láser, en etapas avanzadas solo se pueden gestionar sus efectos y, en casos graves, puede conducir a la ceguera.

La combinación de diabetes tipo 2 y RD representa una carga significativa tanto para la salud de los pacientes como para los recursos económicos del sistema de atención médica y de los propios pacientes. En el campo de la medicina, se utiliza la *electrofisiología* para estudiar la actividad eléctrica en los tejidos, basada en la bioelectricidad generada por la retina y la corteza visual. Una de las técnicas empleadas en esta disciplina es el *electrorretinograma* (ERG), un examen especial que, en este caso, mide el potencial de acción retiniano sin algún estímulo luminoso, utilizando un electrodo corneal y electrodos neutros colocados sobre la piel para medir así las corrientes eléctricas que se generan en la retina.

En la práctica clínica actual, la detección de la retinopatía diabética depende en gran medida del análisis de imágenes del fondo de retina, un procedimiento que requiere equipo especializado, personal capacitado y un costo que no siempre es accesible para todas las clínicas ni para todos los pacientes. Esta dependencia limita la capacidad de realizar diagnósticos tempranos de forma masiva, particularmente en entornos con infraestructura médica restringida. Aunque los electrorretinogramas representan una alternativa menos invasiva para obtener información sobre el estado de la retina, por sí solos no proporcionan un diagnóstico automatizado: las señales obtenidas requieren ser procesadas e interpretadas mediante herramientas computacionales que permitan identificar patrones asociados a la enfermedad.

Esta variedad de corrientes eléctricas se expresa en términos de frecuencia, y siguiendo la misma línea del proyecto colaborativo mencionado al final de la introducción, se les pueden aplicar diversas transformadas matemáticas para obtener información útil de estas señales para que los modelos de AA logren encontrar patrones que indiquen principios de RD. La familia de transformadas utilizada es la *Wavelet*, que incluye tipos como *Shannon*, *Gaussiana Real*, *Morlet Compleja*, entre otros. Las frecuencias obtenidas a partir de estas transformaciones se analizan mediante algoritmos de aprendizaje

automatizado, considerando dos técnicas principales para este trabajo.

Los datos utilizados en este trabajo provienen de un estudio clínico realizado en cuatro instituciones médicas mexicanas: el Instituto Mexicano de Oftalmología (IMO) en Querétaro, el Instituto de la Retina del Bajío (INDEREB) en Querétaro, la Asociación Para Evitar la Ceguera (APEC) en la Ciudad de México y la Clínica de Salud Visual (CSV) de la ENES León, UNAM, en León. Cabe señalar que, por cuestiones de privacidad, no se dispone de información detallada sobre las características demográficas de los participantes, lo cual limita la posibilidad de incorporar estos factores como variables predictivas en los modelos. El conjunto de datos resultante presenta un desbalance entre clases, con una proporción significativamente mayor de pacientes enfermos que sanos (por ejemplo, 22 enfermos frente a 10 sanos), una condición que, como se describió anteriormente, es representativa de los desafíos frecuentes en el ámbito clínico.

Cabe señalar que el proceso de generación de datos sintéticos, llevado a cabo por los responsables del proyecto colaborativo, consideró criterios de balanceo con respecto a variables demográficas como la edad y el sexo de los participantes, con el fin de obtener un conjunto sintético más representativo de la población. No obstante, dado que esta información no estuvo disponible para los fines de esta tesis por las restricciones de privacidad mencionadas anteriormente, dichas variables no fueron incorporadas como predictoras en ninguno de los modelos desarrollados.

Ambas técnicas parten de una limitación compartida: el conjunto de datos disponible para este proyecto era reducido y desbalanceado, con una mayor proporción de pacientes enfermos que sanos. Esta situación es representativa de un problema ampliamente reconocido en el ámbito clínico, donde la obtención de conjuntos de datos suficientes en cantidad y calidad para el entrenamiento adecuado de modelos de aprendizaje automatizado, tanto tradicionales como de aprendizaje profundo, constituye un desafío significativo [7, 8]. Ante esta restricción, cada técnica propone una estrategia distinta para enriquecer la información disponible y mejorar el desempeño de los modelos de clasificación.

La primera técnica consiste en generar imágenes de escalogramas (2D) a partir de una transformada continua de wavelet, lo que permite ampliar la dimensionalidad de la información obtenida mediante los estudios de electroretinogramas (ERG), integrando el tiempo, la escala o frecuencia y la intensidad de la señal en una representación bidimensional. Este enfoque explora la aplicación de *Redes Neuronales Convolucionales* (RNC) con una arquitectura de mayor profundidad, orientada al procesamiento de escalogramas, aprovechando la capacidad de estos modelos para el análisis y clasificación de imágenes, como se detalla en el marco teórico.

Por otro lado, la segunda técnica —complementaria a la anterior— aborda la limitación de los datos desde un ángulo distinto: la generación de información sintética basada en los datos de cada paciente, con el objetivo de incrementar el volumen del conjunto de entrenamiento. La creación de datos sintéticos constituye un área de investigación en sí misma; por ello, en esta tesis se abordará únicamente el principio de su generación dentro del marco teórico. A diferencia del primer enfoque, esta técnica adopta una perspectiva exploratoria más amplia: mediante un sondeo general, se evalúa y compara el desempeño de diversos algoritmos de *Aprendizaje Automatizado* (AA) y de *Aprendizaje Profundo* (AP) con arquitecturas simples en la clasificación de pacientes, evaluando el impacto de la adición de datos sintéticos como factor clave en los resultados.

En este contexto, la relevancia de esta tesis radica en proponer una alternativa accesible a los métodos tradicionales de detección de RD mediante técnicas de AA y AP. Como se mencionó anteriormente,

la primera técnica se orienta hacia la aplicación de *Redes Neuronales Convolucionales* (RNC) sobre los escalogramas generados, mientras que la segunda técnica abarca un sondeo más amplio de algoritmos aplicados sobre los datos enriquecidos con información sintética: algoritmos tradicionales de clasificación como *Árboles de Decisión*, *Bosques Aleatorios*, *Regresión Logística*, *Máquinas de Soporte Vectorial* y *Bayes Ingenuo Gaussiano*, así como modelos de aprendizaje profundo con arquitecturas simples, entre ellos el *Perceptrón Multicapa*, las *Redes Neuronales Convolucionales* y las *Redes Neuronales Recurrentes*. Cabe destacar que, independientemente de la técnica y del algoritmo utilizado, el desempeño de los modelos de aprendizaje automatizado depende en gran medida de la configuración de sus hiperparámetros; por ello, todos los experimentos presentados en esta tesis incluyen un proceso cuidadoso de optimización.

En particular, se seleccionó un método específico de optimización de hiperparámetros con el fin de maximizar el desempeño de cada algoritmo en relación con los datos de estudio, fundamentando su elección en la eficacia que ha demostrado para esta tarea. Los pasos establecidos para llevar a cabo este proceso se describen detalladamente en el capítulo de *Experimentación*.

Asimismo, se emplearon técnicas de selección de características, como el *Análisis Discriminante Lineal* (ADL) y el *Análisis de Componentes Principales* (ACP), como parte de los procesos de preprocesamiento y optimización, con el objetivo de garantizar que las representaciones de los datos sean lo más efectivas y adecuadas posible para los modelos utilizados.

Sección 1.2.

Objetivos

1.2.1. Objetivo General

Desarrollar modelos de aprendizaje automatizado para la detección temprana de retinopatía diabética.

1.2.2. Objetivos Particulares

- Implementar modelos de aprendizaje automatizado, incluyendo *Árboles de Decisión*, *Bosques Aleatorios*, *Regresión Logística*, *Máquinas de Soporte Vectorial*, *Bayes Ingenuo Gaussiano*, *Perceptrón Multicapa*, *Redes Neuronales Recurrentes* y *Redes Neuronales Convolucionales*, para la detección de retinopatía diabética utilizando datos provenientes de estudios de electroretinogramas (ERG) mediante *Escalogramas* y *Datos Sintéticos*. En el caso de los escalogramas, considerar exclusivamente *Redes Neuronales Convolucionales* para su análisis y clasificación.
- Optimizar modelos de aprendizaje automatizado mediante un ajuste fino de hiperparámetros e implementar técnicas de selección de variables como el *Análisis de Componentes Principales* (ACP) y el *Análisis Discriminante Lineal* (ADL), comparando su efecto en el desempeño de los modelos bajo distintas configuraciones experimentales.
- Validar y comparar el desempeño de modelos de aprendizaje automatizado en la clasificación de

pacientes que sean catalogados como sanos o enfermos, considerando el impacto de la cantidad de datos disponibles y las técnicas de preprocesamiento aplicadas, a través de métricas de evaluación como la precisión, exactitud, sensibilidad, especificidad, medida F1 y curva ROC-AUC mediante de mapas de calor que faciliten su representación.

Sección 1.3.

Trabajos Relacionados

En esta sección se busca realizar una exploración a detalle de la literatura existente en el campo de la salud y, específicamente hablando, en la detección de retinopatía diabética (RD) mediante la aplicación de aprendizaje automatizado. Así pues, se revisan las soluciones pioneras que abordan este tema, avanzando hacia las aplicaciones más recientes en el diagnóstico de esta seria enfermedad ocular. Al mismo tiempo, se aborda la identificación de las brechas que persisten en la literatura actual, destacando la necesidad de continuar investigando y desarrollando nuevas técnicas para abordar eficazmente este tema. El presente análisis no solo proporciona una visión retrospectiva de la evolución de la detección de la RD, sino que también subraya la importancia permanente de la innovación y la investigación en este campo que se encuentra en constante cambio.

Antes de proceder con la revisión de los trabajos, es conveniente introducir brevemente las métricas de evaluación que se mencionan en los estudios revisados, las cuales se definen con mayor detalle en la sección 2.6 del marco teórico. La *sensibilidad* indica la proporción de casos positivos (por ejemplo, pacientes enfermos) que el modelo identifica correctamente; la *especificidad* mide la proporción de casos negativos (pacientes sanos) correctamente clasificados; la *precisión* refleja qué tan confiables son las predicciones positivas del modelo; la *exactitud* representa la proporción total de predicciones correctas; y la curva *ROC-AUC* cuantifica la capacidad global del modelo para distinguir entre clases, donde un valor cercano a 1 indica un desempeño excelente.

Comenzamos explorando el trabajo de Ege B. M. et al. realizado en el año 2000 [9]. Este estudio propuso una herramienta innovadora para realizar análisis automatizado de imágenes digitales obtenidas durante el monitoreo rutinario de la retinopatía diabética (RD) en su clínica. La herramienta empleaba un clasificador estadístico elegido entre el clasificador bayesiano, el clasificador de Mahalanobis y el clasificador K-Vecinos más Cercanos (KNN, por sus siglas en inglés). Cada uno de estos clasificadores fue evaluado utilizando un conjunto de 134 imágenes retinales. Los resultados destacaron el rendimiento superior del clasificador de Mahalanobis, logrando una sensibilidad del 69 %, 83 %, 99 %, y 80 % en la detección de microaneurismas, hemorragias, exudados y manchas, respectivamente.

En el trabajo de Osareh A. et al de 2001 [10], se introdujo un método automatizado para la detección de regiones exudadas que utilizó la normalización del color de la imagen para mejorar el contraste entre los objetivos y el fondo. La función de esta normalización era segmentar la imagen de color de la retina en regiones homogéneas utilizando la técnica de agrupamiento de Fuzzy C-Means para clasificar las regiones en exudados y parches no exudados mediante una red neuronal. Los resultados que obtuvieron fueron prometedores, pues alcanzaron una sensibilidad del 92 % y una especificidad del 82 %.

Larsen et al. desarrollaron en 2003 un algoritmo de análisis de imágenes de fondo de retina para la detección automatizada de hemorragias y microaneurismas en pacientes con RD [11]. El estudio evaluó

su rendimiento mediante el análisis de 400 fotografías de fondos de retina con diabetes, seleccionadas al azar del *Welsh Community Diabetic Retinopathy Study*. El algoritmo exhibió una especificidad del 71.4 % y una sensibilidad del 96.7 % en la detección de RD cuando se aplicó un umbral tentativo para su uso en dicha detección. Además, logró una precisión del 79 %. La concordancia de la detección automatizada con el grado visual general (0.659) resultó muy comparable a la concordancia media obtenida por seis oftalmólogos (0.648).

Narasimha-Iyer et al. llevaron a cabo un estudio en 2006 [12], en el cual propusieron un enfoque automatizado para la detección y clasificación de cambios en series temporales longitudinales de imágenes a color del fondo de retina asociadas a la RD. Su método demostró robustez frente a variaciones espaciales en la iluminación, partículas de polvo, valores atípicos y errores de segmentación y alineación. Para alcanzar un rendimiento óptimo frente a estos desafíos, introdujeron algoritmos iterativos innovadores y diversas técnicas de explotación de características espectrales y transformaciones de 12 parámetros. El resultado de su trabajo se traduce en algoritmos bayesianos para la detección y clasificación, generando una salida codificada por colores. La validación se realizó mediante la revisión independiente de tres especialistas sobre 43 pares de imágenes de 22 ojos con RD proliferativa y no proliferativa, alcanzando una tasa de detección de cambios del 97 % y una exactitud en la clasificación de los cambios detectados del 99.3 %.

En el año 2007, Niemeijer et al. [13] desarrollaron un sistema automatizado basado en AA con el propósito de detectar exudados y manchas de algodón en fotografías de fondos de retina de color digitales, y distinguirlos de las drusas para realizar un diagnóstico temprano de la RD. Para este estudio, se extrajeron trescientas imágenes retinianas de un grupo de 300 pacientes con diabetes de una base de datos de telediagnóstico de RD. Este conjunto de datos se utilizó tanto para el entrenamiento como para la evaluación del modelo propuesto. Los resultados mostraron un área bajo la curva ROC de 0,95, así como pares de sensibilidad/especificidad de 0,95/0,88 para la detección de lesiones brillantes de cualquier tipo. Además, se obtuvieron valores de 0,95/0,86, 0,70/0,93 y 0,77/0,88 para la detección de exudados, manchas de algodón y drusas, respectivamente.

Karegowda A.G. et al. llevaron a cabo en 2011 la detección de exudados en imágenes de retina utilizando una red neuronal de propagación hacia atrás (BPN por sus siglas en inglés) [14]. En este estudio, se empleó el conjunto de datos de retinopatía diabética *DIARETDB1*, de dominio público, en el proceso de evaluación. La metodología propuesta incluyó la eliminación del disco óptico para prevenir interferencias en la detección de exudados. Además, se llevó a cabo la identificación de características significativas (reducción de dimensionalidad) después del preprocesamiento de las imágenes mediante los métodos de Árboles de Decisión y Relación de Ganancia, y selección de características basada en correlación (GA-CFS por sus siglas en inglés). Los resultados que se muestran en este trabajo denotan un rendimiento notable de la BPN cuando se utilizan las características identificadas por el árbol de decisiones y el enfoque GA-CFS en comparación con el uso de todas las variables. Dicho clasificador BPN obtuvo su mejor rendimiento con una sensibilidad del 96,97 %, una especificidad del 100 %, y una precisión del 98,45 %.

Ahora bien, De la Calleja J. et al. [15] propusieron en 2014 un método de dos etapas para la detección de RD a partir del análisis de imágenes de fondo de retina. En primer lugar, ellos utilizaron patrones binarios locales (LBP) como método de extracción de características. Posteriormente, aplicaron redes neuronales artificiales, bosques aleatorios y máquinas de soporte vectorial para detectar la RD, a través de un conjunto de datos compuesto por 71 imágenes. Los resultados obtenidos mostraron que el algoritmo

de Bosques Aleatorios fue el que mejor desempeño obtuvo con una precisión del 97.46 %.

En el artículo de Roychowdhury, S. et al del 2014 [16], se propone un sistema de detección asistida por computadora que analiza imágenes del fondo del ojo para generar un grado de gravedad de la RD mediante técnicas de aprendizaje automatizado. El enfoque incluye una clasificación jerárquica mediante el uso de Adaboost, lo que resulta en mejoras significativas en la precisión en comparación con la literatura previa. Además, este enfoque logra reducir de manera considerable el costo computacional, lo que es una ventaja significativa para su implementación práctica.

En el estudio realizado por Hagos M.T. et al. en 2019 [17], se aborda la problemática inherente de disponer de un conjunto de datos limitado en el contexto clínico. Ante esta limitación, proponen la aplicación del *Aprendizaje por Transferencia*, que se basa en la utilización de una estructura de red neuronal previamente entrenada con un conjunto de datos masivo, no necesariamente vinculado directamente al objeto de estudio. En este contexto, optaron por emplear la red neuronal profunda del modelo *Inception-V3*, aprovechando sus módulos Inception para la detección de retinopatía diabética y superando la escasez de datos. Los resultados obtenidos del análisis de imágenes mediante este método son notables, pues cuentan con una precisión del 90.9 %.

En un estudio previo de Reddy G.T. et al en 2020, [18] adoptaron un enfoque de *Ensemble Learning* o Aprendizaje por Conjuntos para abordar la detección de la RD. La propuesta lleva a cabo una comparación exhaustiva de los resultados obtenidos por varios algoritmos de aprendizaje automatizado, incluyendo Árboles de Decisión, K-Vecinos más Próximos (algoritmo que clasifica una observación con base en la clase mayoritaria de sus K vecinos más cercanos en el espacio de características; *K-Nearest Neighbors* en inglés), Bosques Aleatorios, Adaboost y Regresión Logística. A través del método de Búsqueda por Malla (método que evalúa exhaustivamente todas las combinaciones posibles de un conjunto predefinido de hiperparámetros; *Grid Search* en inglés), se optimizan los hiperparámetros de los clasificadores. Finalmente, se comparan las clasificaciones de cada algoritmo optimizado para concluir con el resultado que más se repite entre los clasificadores. Este estudio demostró que este enfoque supera los resultados individuales de estos algoritmos, alcanzando una exactitud del 78 %. Estos hallazgos resaltan una tendencia emergente en el campo, donde las estrategias de Aprendizaje por Conjuntos están demostrando ser efectivas para mejorar el rendimiento en la detección de la RD.

En uno de los trabajos más recientes del año 2021, Khan et al. [19] llevaron a cabo una investigación centrada en la clasificación de las diferentes etapas de la RD, utilizando parámetros mínimos para acelerar tanto la formación como la convergencia del modelo. El modelo empleado fue el VGG-16, complementado con una capa de agrupación de pirámides espaciales (SSP, por sus siglas en inglés) y la Red en Red (NiN, por sus siglas en inglés), dando lugar a un modelo profundo a escala invariante altamente no lineal denominado VGG-NiN. Este modelo demostró su capacidad para procesar imágenes de RD a cualquier escala, mostrando mejoras significativas en la clasificación. Los resultados experimentales revelaron que el modelo propuesto exhibe un rendimiento superior en términos de exactitud, alcanzando un 85 %, y al mismo tiempo logra una utilización más eficiente y menos costosa de los recursos computacionales.

Después de realizar este análisis de literatura para esta problemática, se logra identificar una evolución notable en la detección de la retinopatía diabética (RD) mediante el aprendizaje automatizado. No obstante, se observa que los trabajos revisados se enfocan predominantemente en la mejora del desempeño algorítmico, dejando de lado la cuestión de ofrecer alternativas de detección que sean más accesibles y menos dependientes de equipos especializados o costosos. Este aspecto resulta relevante si se considera que una solución práctica para la detección temprana de la RD debería ser viable tanto en términos de

precisión como de facilidad de implementación en entornos clínicos con recursos limitados.

Estas lagunas subrayan la necesidad urgente de investigaciones adicionales y métodos innovadores que aborden estos desafíos específicos. Aunque los trabajos revisados allanan el camino para comprender la RD desde una perspectiva algorítmica y visual, el desafío de detectar la RD en su etapa más temprana con métodos no intrusivos y fiables sigue siendo un terreno sin explorar.

En síntesis, esta revisión sirve como un resumen de los logros existentes, además de apoyar a un llamado a la acción para la comunidad científica con la finalidad de abordar las brechas identificadas. Ahora bien, la sección que continúa en este trabajo se enfoca en sentar las bases teóricas para un correcto entendimiento de la problemática abordada para que, posteriormente, se logre concretar la metodología propuesta para cumplir con los objetivos planteados.

Marco Teórico

El análisis de la literatura enfatiza la necesidad urgente de abordar las brechas existentes en la detección de la retinopatía diabética. La sección subsiguiente de esta tesis se centra en el “Marco Teórico”, que establece las bases conceptuales y metodológicas esenciales para comprender de manera profunda la problemática que se plantea.

Durante el recorrido de los capítulos se exploran conceptos clave como los escalogramas, la inteligencia artificial, el aprendizaje automatizado, el aprendizaje profundo, la extracción de características y las métricas de evaluación. Dentro del AA y del AP se abordarán los conceptos básicos de algunos algoritmos tradicionales más distintos tipos de redes neuronales, así como la generación de datos sintéticos. Todo esto con la finalidad de construir los fundamentos sólidos para lograr comprender de forma integra la fase de experimentación que conlleva este trabajo.

Sección 2.1.

Escalogramas

Un concepto fundamental que se debe abordar antes de pasar a los conceptos de inteligencia artificial y sus involucrados, son los *Escalogramas*. Según Scarpiniti, M. et al. (2023) [20] un escalograma se puede definir como el módulo al cuadrado de la Transformada Continua de Wavelet (CWT por sus siglas en inglés), la cuál es una representación del tiempo, la frecuencia y la intensidad de una señal. También menciona que es una herramienta poderosa para analizar problemas del mundo real que involucren señales.

Una Wavelet se refiere a un tipo de transformación matemática que convierte una señal que está en el dominio del tiempo al dominio de tiempo-escala, es decir, una representación matricial de la señal donde cada fila corresponde a una escala y cada columna a un instante en el tiempo. Esto permite analizar señales complejas que involucran una variación en la frecuencia con respecto al tiempo, es decir, señales no estacionarias. Una de sus ventajas es que permite analizar tanto frecuencias altas como frecuencias bajas, además de que permite examinar diferentes partes de la señal de forma matemática.

Algunos de los tipos de familias de Wavelet son la *Wavelet Sobrero Mexicano*, *Wavelet Morse*, *Wavelet Bump* o la *Wavelet Morlet* que tiene dos variantes, la *Wavelet Morlet Real* y la *Wavelet Morlet Compleja*.

La forma de obtener un escalograma es a través de la Transformada Continua de Wavelet (TCW)

que utiliza alguna familia de Wavelet como las mencionadas en el párrafo anterior, según el criterio que se elige para un análisis específico. Cuando se le aplica la TCW, se obtiene una matriz de coeficientes que, dado que estos coeficientes pueden ser complejos, se elevan al cuadrado para que, posteriormente, se puedan graficar en un mapa de calor y obtener un escalograma que represente el tiempo en el eje X, la frecuencia en el eje Y y la intensidad de la señal con el color de la imagen.

En este sentido, dado que los escalogramas demuestran resultados efectivos en diversos enfoques que van desde la clasificación del sonido ambiental hasta el estudio de las señales de electroencefalogramas, se complementa el análisis del mismo a través de algoritmos complejos de Redes Neuronales Convolucionales que serán definidas más adelante.

Sección 2.2.

Inteligencia Artificial

La *Inteligencia Artificial* es un campo vasto y en constante evolución, con debates persistentes sobre su definición precisa. En el libro titulado en inglés *Artificial Intelligence: A Modern Approach*, J. Russell et al. profundizan en el concepto, su origen y la historia de la IA. Personalmente, encuentro la definición de Bellman (1978) bastante acertada; parafraseando sus palabras, la IA es la automatización de actividades que asociamos con el pensamiento humano, como la toma de decisiones, la resolución de problemas y el aprendizaje a partir de situaciones particulares [21].

A lo largo del tiempo, se considera un renombramiento de lo que hoy llamamos *Inteligencia Artificial*, ya que, desde un punto de vista crítico, en la actualidad este término se utiliza más como una estrategia de marketing que como una definición de nuevos métodos o herramientas que contribuyen al desarrollo continuo del campo. Lo que sí podemos afirmar es que la IA es un área de estudio que abarca varios subcampos interrelacionados. Algunos de estos subcampos incluyen Ciencia de Datos, Aprendizaje Automatizado, Aprendizaje Profundo, Minería de Datos, Procesamiento de Lenguaje Natural, entre otros.

Las áreas de la IA de alta relevancia para esta tesis son el *Aprendizaje Automatizado* (AA) y el *Aprendizaje Profundo* (AP). La selección de los algoritmos explorados en ambas áreas responde a las características particulares del problema abordado: un conjunto de datos originalmente reducido, desbalanceado y proveniente de señales biomédicas. Si bien uno de los enfoques del trabajo consistió en enriquecer dicho conjunto mediante la generación de datos sintéticos, la selección de modelos se realizó considerando su pertinencia tanto para condiciones de datos limitados como para escenarios con mayor volumen de información, con el fin de evaluar comparativamente su desempeño bajo distintas condiciones experimentales. En este sentido, los modelos tradicionales de AA —como bosques aleatorios, regresión logística, bayes ingenuo gaussiano, máquinas de soporte vectorial y árboles de decisión— fueron considerados por su robustez y su capacidad para ofrecer resultados interpretables, lo cual resulta relevante en contextos clínicos. Los modelos de AP —perceptrón multicapa, redes neuronales recurrentes y redes neuronales convolucionales— fueron incorporados para explorar si arquitecturas más complejas presentan ventajas sobre los métodos tradicionales bajo las mismas condiciones de datos. Ambas áreas se definen y describen en las secciones siguientes.

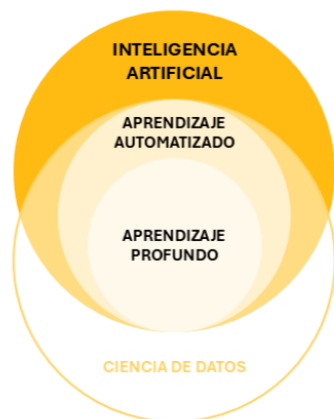


Figura 2.1: Áreas relacionadas a la IA.
Imagen Castellanzada de Oracle, [22]

Sección 2.3.

Aprendizaje Automatizado

Tal y como se menciona en la sección anterior, el Aprendizaje Automatizado (AA) es un subcampo fundamental dentro del marco de la Inteligencia Artificial que, según plantea [23], se ha ganado una relevancia muy significativa en los últimos años. Dentro de la tecnología moderna, el AA surge como un pilar principal puesto que impulsa avances en distintas disciplinas las cuáles ayudan a transformar aspectos clave de nuestra vida diaria. En este sentido, aquí se define el concepto de Aprendizaje Automatizado, sus principales subáreas y se profundiza en los conceptos fundamentales de los algoritmos de Aprendizaje Automatizado que se utilizan en esta tesis.

De acuerdo con Alpaydin, E. (2020) [2], el aprendizaje automatizado se puede entender como la programación de computadoras para optimizar un criterio que permite evaluar el rendimiento obtenido en relación con una tarea específica. Alpaydin explica que los modelos están definidos por parámetros internos, y para generar *aprendizaje*, se busca que el programa informático optimice dichos parámetros utilizando datos de ejemplo (experiencia pasada).

Estos modelos pueden emplearse tanto para hacer predicciones futuras (modelos predictivos), como para extraer conocimiento de los datos (modelos descriptivos), o incluso ambos. Sin embargo, generalmente se reconoce que existen tres tipos principales de aprendizaje automatizado: supervisado, no supervisado y aprendizaje por refuerzo. Para comprender estas categorías, es importante mencionar que, en el contexto del aprendizaje automatizado, una *etiqueta* se refiere a la señal de salida deseada o conocida que se asocia a cada observación en un conjunto de datos, por ejemplo, si un paciente está sano o enfermo.

En su libro *Python Machine Learning*, Raschka, S. et al. (2019) [24], abordan la definición de estas tres categorías. Comenzando con el aprendizaje supervisado, los autores lo definen como un proceso en el cual un modelo aprende a partir de datos de entrenamiento *etiquetados*, y gracias a estas etiquetas es

posible realizar predicciones sobre datos no vistos o futuros. El término “supervisado” hace referencia a los conjuntos de datos de entrenamiento en los que las señales de salida deseadas (etiquetas) ya son conocidas.

Por otro lado, el aprendizaje no supervisado se caracteriza por la ausencia de datos etiquetados. A diferencia del enfoque anterior, el objetivo en esta categoría no es predecir nuevos datos a partir de entradas previas, sino identificar patrones o similitudes en los datos, agrupándolos en conjuntos que compartan características comunes, todo ello basado en criterios matemáticos.

Además, ellos explican que la rama del AA, aprendizaje por refuerzo, se refiere al proceso computacional que mejora su rendimiento mediante la interacción con el entorno. A este sistema le llaman *Agente*, el cual recibe una *señal de recompensa* durante el proceso de mejora para evaluar si el agente está logrando el objetivo deseado. Entre más señales de recompensa recibe, más ajusta su comportamiento a través del método de ensayo y error en donde, en un principio sigue estrategias aleatorias hasta que llega a encontrar un patrón que maximice las recompensas y optimice su desempeño.

Además, los autores exploran la *clasificación* como una subcategoría del aprendizaje supervisado, cuyo objetivo es predecir las etiquetas de clase categóricas de nuevas instancias basándose en observaciones pasadas. Es decir, en lugar de predecir un valor continuo como en la regresión, se busca clasificar una observación dentro de categorías previamente definidas. Existen dos tipos de clasificación: la clasificación binaria, que comprende la predicción de una etiqueta para dos valores posibles (por ejemplo, verdadero/falso o 1/0), y la clasificación multiclase, que abarca la predicción de múltiples categorías para clasificar una observación (por ejemplo, 1, 2, 3, 4).

Algunos de los algoritmos más conocidos dentro del aprendizaje supervisado, tanto para tareas de regresión como de clasificación, incluyen la regresión lineal, la regresión logística, los árboles de decisión, y el algoritmo de bayes ingenuo, entre otros. No obstante, para esta tesis, nos enfocaremos en profundizar en el algoritmo de *Bosques Aleatorios*, el cual se explorará con mayor detalle en la siguiente sección.

2.3.1. Árboles de Decisión y Bosques Aleatorios

Los árboles de decisión son un método basado en la descomposición jerárquica de un problema, en el que una pregunta compleja se divide sucesivamente en preguntas más simples hasta llegar a una decisión final.

Autores como Alpaydin, E. (2020 [2], Russell, S., Norvig, P (2016) [21], Rebala, G., Ravi, A., Churiwala, S. (2019) [23] y Raschka, S. y Mirjalili, V. (2019) [24] concluyen que los árboles de decisión son uno de los algoritmos más utilizados en el aprendizaje automatizado supervisado. Su versatilidad radica en su capacidad para abordar tanto problemas de regresión como de clasificación, ya sea binaria o múltiple. Además, estos árboles pueden manejar datos numéricos y categóricos, y su alta interpretabilidad facilita la comprensión de los resultados. Así, las decisiones tomadas a lo largo del árbol pueden expresarse como un conjunto de reglas del tipo *si-entonces*, cuyos componentes se describen más adelante.

Un árbol de decisión se enfoca principalmente en construir una estructura jerárquica que divide el conjunto de datos en conjuntos más pequeños con base en condiciones específicas. Los autores citados en el párrafo anterior plantean que este algoritmo aplica el principio de *divide y vencerás*, debido a que

el algoritmo comienza a buscar puntos de decisión que simplifiquen el problema en partes más sencillas.

La estructura general de un árbol de decisión es la siguiente:

- **Nodo principal:** También conocido como *nodo raíz*, representa la totalidad del conjunto de datos y se dividirá con base en un criterio.
- **Nodo de decisión:** Son los puntos en los que los datos comienzan a dividirse, dependiendo de una decisión o umbral específico. El nodo raíz también se considera un nodo de decisión.
- **Nodo hoja:** Aquí se toma la decisión final y se obtiene el resultado.

Cabe mencionar que la profundidad de un árbol indica cuántos niveles tiene hacia abajo, mientras que cada nivel se define con al menos un nodo de decisión. Por ejemplo, un árbol con una profundidad de 5 quiere decir que su primer nivel es el nodo principal, los niveles 2, 3 y 4 tienen nodos de decisión y el último nivel tiene nodos hoja.

El árbol de decisión está compuesto por nodos internos y nodos hoja. Cada nodo de decisión aplica una función de prueba $f_m(x)$, la cual determina hacia qué rama se debe dirigir la instancia según el resultado obtenido. Este proceso continúa desde el nodo raíz hasta llegar a un nodo hoja, que contiene la predicción o el valor final.

La función de prueba puede surgir de diferentes modelos dependiendo del método utilizado en el árbol de decisión. Además, cada uno de los nodos hoja genera una salida que, para el caso de clasificación, es la clase objetivo, mientras que para la regresión, es un valor de la variable a predecir.

Debido a la reducción de reglas del tipo *si-entonces*, cuando la entrada del árbol es univariable, para el caso de valores continuos, se considera la siguiente ecuación:

$$f_m(x) : x_j > w_{m0} \quad (2.1)$$

Donde, de acuerdo con Alpaydin, E. (2020) [2]:

- x_j : significa la dimensión de la entrada utilizada.
- w_{m0} : es el valor escogido como umbral.

Aunque la comparación se basa en si sobrepasa un umbral o no, en el caso de los valores discretos, el nodo evalúa el valor de x_j y realiza una división en n subconjuntos, donde n representa el número de valores distintos en el conjunto x_j .

Cuando se llega a un nodo, este algoritmo se enfoca en encontrar un criterio de decisión basándose en la mejor división en el árbol. Cabe mencionar que esta selección dependerá del tipo de dimensión de entrada utilizada y la forma de identificar el criterio de decisión adecuado, este algoritmo implementa un concepto llamado *Ganancia de Información*. Esta *Ganancia de Información* puede interpretarse como una medida cuantitativa del valor que aporta algún criterio de división. Es decir, si el objeto de estudio es determinar si lloverá o no (una decisión binaria), y se cuentan con variables como la humedad relativa

del día anterior y la temperatura del día anterior. Podemos suponer que la humedad relativa del día anterior y la temperatura son variables relacionadas con el objeto de estudio, sin embargo, muchas veces se encuentra la problemática de no saber cuál variable explica mejor el problema. Es aquí donde este concepto toma relevancia, pues te ayuda a saber cuál variable es más explicativa con base en el valor de la entropía de cada una de ellas.

En el caso de un árbol de decisión para clasificación, la calidad de una división se cuantifica a través de una medida de *impureza*. Una división es considerada pura si, después de la partición, todas las instancias que siguen una rama pertenecen a la misma clase. Para un nodo m , el número de instancias de entrenamiento que alcanzan dicho nodo es N_m . En el nodo raíz, este valor es N . Según Alpaydin, E. (2020) [2], N_m^i representa el número de instancias en el nodo m que pertenecen a la clase C_i , cumpliendo que $\sum_i N_m^i = N_m$. La probabilidad estimada de que una instancia en el nodo m pertenezca a la clase C_i se expresa como:

$$\hat{P}(C_i|x, m) \equiv (p^i)_m = \frac{N_m^i}{N_m}.$$

Un nodo es puro si $(p^i)_m$, para todas las clases i , es igual a 0 o 1, lo que conlleva a que todas las instancias del nodo pertenezcan a una sola clase. En este caso donde el nodo es puro, ya no es necesario continuar con más divisiones y se puede etiquetar como un nodo hoja con la clase correspondiente.

Una función común para medir la impureza es la *Entropía*, citada por Alpaydin, E. (2020) [2] (basado en Quinlan, 1986), y se define como:

$$I_m = - \sum_{i=1}^k (p_m^i \log_2(p_m^i)),$$

donde $0 \log 0 \equiv 0$. En teoría de la información, la entropía especifica el número mínimo de bits necesarios para codificar la clase de una instancia. En un problema binario, si $p^1 = 1$ y $p^2 = 0$, todas las instancias pertenecen a la clase C^1 , y no se requiere transmitir ningún bit de información, lo que implica que la entropía es 0. Si $p^1 = p^2 = 0.5$, entonces se necesita un bit para identificar a cuál de las dos clases pertenece la instancia, lo que da como resultado una entropía de 1. En situaciones intermedias, es posible utilizar códigos más eficientes, asignando códigos más cortos a la clase más probable y códigos más largos a la clase menos probable. Cuando hay $K > 2$ clases, el mismo principio aplica, y la mayor entropía posible es $\log_2 K$, cuando $p^i = 1/K$ para todas las clases i .

En el contexto de los árboles de decisión, la entropía se utiliza como medida de impureza en un nodo. Un nodo con entropía baja está compuesto mayormente por instancias de una sola clase, mientras que un nodo con alta entropía tiene una distribución más balanceada entre varias clases. El objetivo en los árboles de decisión es reducir la entropía con cada división, logrando nodos más puros que permitan clasificaciones más precisas. Por ejemplo, en un nodo con 100 instancias, donde 60 pertenecen a la clase “gato”, 30 a “perro” y 10 a “pájaro”, las probabilidades para cada clase serían $p_m^{\text{gato}} = 0.6$, $p_m^{\text{perro}} = 0.3$ y $p_m^{\text{pájaro}} = 0.1$. La entropía de este nodo sería:

$$I_m = -(0.6 \log_2 0.6 + 0.3 \log_2 0.3 + 0.1 \log_2 0.1),$$

y el valor resultante reflejaría el grado de impureza. Un nodo puro, donde todas las instancias

pertenecen a una sola clase, tendría una entropía de 0, lo que indica que no hay desorden y no se requieren más divisiones.

De lo anterior, surge la métrica conocida como *ganancia de información*, que se define como la diferencia de entropía de un conjunto de datos antes y después de realizar una partición. Esta métrica cuantifica cuánto mejora la pureza de los nodos tras una división. Formalmente, la ganancia de información se define como la diferencia entre la entropía antes de la división y la entropía ponderada de los nodos hijos después de la partición.

Dada una partición del conjunto de datos en t subconjuntos, la ganancia de información IG para un nodo m al dividir en estos subconjuntos se expresa como:

$$IG = I_m - \sum_{j=1}^t \frac{N_j}{N_m} I_j,$$

donde I_m es la entropía del nodo original m , I_j es la entropía de cada nodo hijo j , N_j es el número de instancias en el nodo hijo j , y N_m es el número total de instancias en el nodo original.

El objetivo de esta fórmula es mostrar que, al realizar una partición en el nodo, se reduce la incertidumbre (es decir, la entropía), y la ganancia de información mide dicha reducción. Cuanto mayor sea la ganancia de información, más efectiva será la partición en términos de clasificación de los datos.

Por ejemplo, supongamos que tenemos un nodo m con una entropía inicial de $I_m = 0.971$. Si lo dividimos en dos nodos hijos, donde uno tiene una entropía de $I_1 = 0.811$ y contiene el 70% de las instancias, mientras que el otro tiene una entropía de $I_2 = 0.5$ y contiene el 30% de las instancias, la ganancia de información se calcularía como:

$$IG = 0.971 - (0.7 \cdot 0.811 + 0.3 \cdot 0.5) = 0.971 - 0.748 = 0.223.$$

Esto indica que la partición redujo la impureza del nodo, lo que refleja una mejora en la capacidad de clasificación del árbol en ese punto de decisión.

Como se puede observar, la ganancia de información utiliza a la entropía para maximizar la reducción de la misma en cada etapa del árbol de decisión. Esto lo lleva a cabo al seleccionar el atributo con mayor ganancia de información para que el árbol realice las divisiones que permitan clasificar de una forma más eficiente, a medida que profundiza en las particiones de los nodos.

El algoritmo general para construir un árbol de decisión sigue los siguientes pasos:

Para concluir, el concepto de *bosques aleatorios* no es más que una colección de árboles de decisión en los que cada árbol individual evalúa una serie de decisiones comenzando desde el nodo raíz y progresando por los nodos que cumplen con las condiciones especificadas. Este proceso continúa hasta alcanzar un nodo hoja, el cual contiene la predicción o el resultado de ese árbol. Cada árbol en el bosque puede generar diferentes particiones y, por ende, obtener resultados distintos. La forma en que se obtiene una salida final en un bosque aleatorio de clasificación es a través de un *voto*; es decir, la predicción final será aquella que se repita con mayor frecuencia entre todas las predicciones de los árboles de decisión.

En resumen, la estructura de los árboles de decisión, combinada con el cálculo de la ganancia de

Algoritmo 1 Construcción de un Árbol de Decisión

Requiere: Conjunto de elementos E .**Garantiza:** Un árbol de decisión DT .

- 1: Inicializar un nodo raíz S tal que $S = E$.
 - 2: **Si** todos los elementos de S pertenecen a la misma clase **entonces**
 - 3: Cerrar el subárbol y retornar la solución.
 - 4: **Si no**
 - 5: Elegir una condición de partición para dividir E , siguiendo un criterio de partición (*split criterion*) en E_1, E_2, \dots, E_n .
 - 6: Subdividir el árbol en subárboles: los que cumplen la condición y los que no.
 - 7: Repetir el paso 5 para cada subárbol generado.
 - 8: **Fin Si**
-

información, proporciona un método interpretable para clasificar datos, dado que el proceso de decisión puede rastrearse paso a paso, a diferencia de otros modelos que operan como cajas negras. Los bosques aleatorios, por su parte, incrementan la robustez del modelo al promediar múltiples árboles, lo que permite obtener predicciones más precisas y reducir el riesgo de sobreajuste. Así, estas técnicas continúan siendo una herramienta relevante para la clasificación y predicción en diversos problemas de aprendizaje automatizado.

2.3.2. Regresión Logística

La regresión logística es uno de los modelos más robustos y ampliamente utilizados en problemas de clasificación binaria, y también se adapta para la clasificación en múltiples clases. La principal razón de su uso extendido se debe a sus sólidas bases estadísticas, y su capacidad para ofrecer métricas de evaluación satisfactorias en escenarios donde las clases son altamente separables, todo esto mientras mantiene bajos requisitos computacionales en comparación con los algoritmos de aprendizaje profundo [23].

Una curiosidad sobre su nombre radica en que, aunque incluye la palabra “Regresión”, este modelo se emplea exclusivamente para clasificar observaciones discretas en lugar de realizar predicciones continuas, lo que en principio podría contradecir su denominación.

La base teórica de este algoritmo descansa en la estimación de probabilidades. Es decir, calcula qué tan probable es que una observación pertenezca a una clase específica, empezando por la construcción de una combinación lineal de las variables de entrada, como se representa en la siguiente ecuación:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (2.2)$$

Donde, conforme lo establece Rejala, G. et al [23]:

- x_i : son las variables independientes o de entrada.
- β_n : se refiere a los coeficientes que representan la contribución de cada variable al modelo.

- β_0 : se puede explicar como el sesgo del modelo.

Los valores óptimos de β_0 y β_n se obtienen mediante algún método de optimización que minimiza una *función de pérdida*. Cabe mencionar que este concepto de función de pérdida será profundizado en el contexto de Redes Neuronales Convolucionales más adelante en el documento.

Luego, para convertir esta combinación lineal en una probabilidad comprendida entre los valores 0 y 1, se aplica una función denominada *Sigmoide* o logística:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$

Una vez que el modelo ha sido entrenado, se establece un umbral (ajustable según las necesidades del problema en cuestión) para determinar la clase a la que pertenece cada observación. Por ejemplo:

$$\sigma(z) \geq 0.5 \rightarrow \text{Clase 1}$$

$$\sigma(z) < 0.5 \rightarrow \text{Clase 0}$$

En síntesis, este algoritmo ofrece ventajas notables, como su facilidad de interpretación, rapidez y eficiencia en problemas que son lineales. Sin embargo, una de sus limitaciones es su incapacidad para manejar problemas no lineales sin una transformación previa de los datos para linealizar la relación. Además, la regresión logística es sensible a valores atípicos, los cuales pueden influir negativamente en el desempeño del modelo.

2.3.3. Bayes Ingenuo Gaussiano

El algoritmo Naive Bayes, al igual que la regresión logística, ha sido ampliamente utilizado en diversos campos debido a los principios estadísticos del Teorema de Bayes. Este teorema permite actualizar las probabilidades asociadas a una hipótesis (como puede ser la pertenencia a una clase) en función de nueva evidencia, representada por los datos de entrada [23].

La fórmula general del Teorema de Bayes describe la probabilidad de que un evento ocurra dado cierto conocimiento a priori. La forma en como se expresa en términos es la siguiente:

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)} \quad (2.4)$$

Donde:

- $P(C_k|x)$: Es la probabilidad posterior de que la muestra x pertenezca a la clase C_k .
- $P(x|C_k)$: Es la verosimilitud, que representa qué tan probable es observar x si se sabe que pertenece a la clase C_k .

- $P(C_k)$: Es la probabilidad previa de la clase C_k antes de observar los datos.
- $P(x)$: Es la probabilidad total de observar x bajo todas las clases posibles.

Dado que $P(x)$ es constante para todas las clases, esta ecuación se puede simplificar de la siguiente manera:

$$P(C_k|x) \propto P(x|C_k) \cdot P(C_k) \quad (2.5)$$

En el caso particular del algoritmo Naive Bayes Gaussiano, la verosimilitud $P(x|C_k)$ de cada característica (x_i) se modela utilizando una distribución normal, definida matemáticamente como:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{k,i}^2}} \cdot e^{-\frac{(x_i-\mu_{k,i})^2}{2\sigma_{k,i}^2}} \quad (2.6)$$

Donde:

- $\mu_{k,i}$: Es la media de la característica x_i para la clase C_k .
- $\sigma_{k,i}^2$: Es la varianza de la característica x_i para la clase C_k .

Además, dado que este algoritmo asume independencia entre las variables, la verosimilitud del conjunto de características se calcula como el producto de la verosimilitud de cada característica individual:

$$P(x|C_k) = \prod_{i=1}^n P(x_i|C_k) \quad (2.7)$$

A pesar de que en problemas del mundo real las suposiciones de independencia y normalidad gaussiana rara vez se cumplen estrictamente, este algoritmo funciona sorprendentemente bien en muchos escenarios. La razón principal radica en su principio de operación, donde las probabilidades individuales de las características se multiplican por la probabilidad previa de la clase una vez obtenidos la media y la varianza. Esta simplicidad le permite determinar la clase con mayor probabilidad posterior de forma efectiva.

Este algoritmo tiene bondades como la rapidez y eficiencia que puede presentar en ciertas aplicaciones, su buen desempeño cuando se aplica a grandes conjuntos de datos o su gran resistencia ante el sobreajuste en problemas donde las clases están claramente separadas. No obstante, cuenta con algunas limitaciones como su incapacidad para capturar relaciones entre variables o su necesidad de que se tome como supuesto que las variables sean independientes.

2.3.4. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM, por sus siglas en inglés) son algoritmos cuya historia se remonta al siglo pasado, pero que continúan siendo fundamentales en problemas de clasificación binaria, ya sea con datos lineales o no lineales. Debido a su robustez y efectividad, su uso se ha extendido tanto que incluso modelos avanzados como el aprendizaje profundo, que se explorará en la próxima sección, las integra como último componente para clasificar elementos [23].

Este algoritmo trabaja con un conjunto de datos que contiene un número definido de características y tiene como objetivo principal la clasificación de observaciones en una o varias clases. La tarea fundamental de las máquinas de soporte vectorial es encontrar una línea, llamada *hiperplano*, que divida de forma óptima a las clases que se buscan clasificar.

El término *hiperplano* proviene del cálculo vectorial y se utiliza como una denominación general; dependiendo de la dimensionalidad del espacio, puede representar desde una línea en dos dimensiones hasta un plano o una estructura más compleja en dimensiones superiores. En otras palabras, el hiperplano es el concepto genérico que define esta separación.

Aunque el hecho de determinar un hiperplano que logre separar las clases no es una tarea titánica, se han aplicado múltiples tipos de hiperplano para separar de una mejor manera los conjuntos caprichosos de información. Aquí es donde se presenta el reto principal de este algoritmo, pues la tarea de identificar el hiperplano que logre esta separación óptima puede tardarse bastante tiempo.

Algunos conceptos clave de este algoritmo se presentan a continuación:

- **Vectores de soporte:** Son los puntos específicos del conjunto de datos (también llamados observaciones) que se encuentran más cercanos al hiperplano decidido. Estos puntos muestran la posición y la orientación de dicho hiperplano.
- **Márgen:** Se refiere a la distancia entre los vectores de soporte y el hiperplano.

Un hiperplano sencillo, cuando los datos son linealmente separables, funciona de una forma muy eficiente. No obstante, en la mayoría de los casos, los datos presentan distribuciones no lineales que agregan tiempo y complejidad a la separación de los mismos. Un método que se implementó para abordar esta limitación fue la utilización de *kernels*. Los *kernels* son funciones que aplican transformaciones matemáticas al conjunto de datos, permitiendo representarlos en un espacio de mayor dimensionalidad donde puede ser más sencillo encontrar un hiperplano que logre una separación óptima. Una vez que el hiperplano ha sido calculado en este espacio transformado, el algoritmo regresa al espacio original, ajustando los datos y el hiperplano encontrado mediante la misma transformación inversa.

Existen diversos tipos de *kernels*, como el lineal, polinómico, gaussiano (función radial base, RBF, por sus siglas en inglés), entre otros. Cada uno de ellos puede adaptarse mejor o peor al patrón de los datos, dependiendo del problema a resolver. Esta capacidad para modelar transformaciones y encontrar hiperplanos óptimos incluso en problemas con estructuras complejas ha consolidado a las máquinas de soporte vectorial como uno de los algoritmos más robustos y versátiles de la clasificación en aprendizaje automatizado, tanto para casos binarios como de múltiples clases.

Sección 2.4.

Aprendizaje Profundo

Al igual que el Aprendizaje Automatizado, el Aprendizaje Profundo (AP) es un subcampo crucial dentro de la Inteligencia Artificial que surge como uno de los temas más innovadores del siglo XXI. Debido a su rápido crecimiento y a su potencial aún en gran parte inexplorado, el Aprendizaje Profundo ofrece numerosas oportunidades para avanzar en el campo de la Inteligencia Artificial. En esta sección, se explora la definición del Aprendizaje Profundo, se presentan algunos ejemplos de sus aplicaciones más destacadas y se profundiza en los conceptos fundamentales de los algoritmos de AP que se utilizan en esta tesis.

En el libro por excelencia sobre AP de Goodfellow et. al. (2016) [25], podemos apreciar que el surgimiento de este nuevo tipo de aprendizaje se comenzó por explorar a través de la búsqueda de replicar el funcionamiento del pensamiento humano. Tocando en términos simples dicho tópico, *grosso modo*, el cerebro recibe una señal de entrada que, a través de ciertos procesos dentro de las neuronas, se interpreta dicha señal y se obtiene una señal de salida que representa la información de dicho estímulo. En este sentido, uno de los primeros algoritmos que se desarrollaron como primeros acercamientos a este aprendizaje fue el *perceptron*. Dicho algoritmo funciona básicamente con una entrada que se transforma a través de una función de activación (función matemática) y se obtiene una salida que se evalúa si es la esperada o no, tal y como lo hace de forma muy simplificada las neuronas del cerebro humano.

En la próxima sección, procederemos a examinar la definición de las redes neuronales artificiales, con especial atención a las redes neuronales de perceptrón multicapa, redes neuronales recurrentes, redes neuronales convolucionales y redes generativas adversarias, dado que son los principales algoritmos de AP de este trabajo.

2.4.1. Redes Neuronales

En consonancia con la idea planteada por Goodfellow en la sección previa y sustentado por la definición del aprendizaje profundo (AP) proporcionada por Roberts (2022) [26], se indica que aunque el AP se basa en el modelo de aprendizaje de redes neuronales biológicas, opera principalmente como un conjunto flexible de funciones construidas sobre numerosos bloques computacionales, considero que el objetivo primordial del AP es fortalecer progresivamente estos bloques. Esto permite procesar y entrenar con datos del mundo real, facilitando el aprendizaje para la resolución efectiva de problemas.

En el ámbito del aprendizaje profundo (AP), existe una variedad de redes neuronales tradicionales, como las mencionadas por Olabe (1998) [27], que incluyen:

- Perceptron Multicapa (*Multilayer Perceptron*).
- Adaline (*Adaptive Linear Neuron*).
- Madaline (*Multi Adaptive Linear Neurons*).
- Redes autoorganizadas (*SOM*, por sus siglas en Inglés), *Hopfield*, *BAM*.

- Redes ART (*Adaptive Resonance Theory*)

Actualmente, se han desarrollado nuevos tipos de redes neuronales, entre las cuales se encuentran:

- Redes Neuronales Recurrentes y subcategorías.
- Redes Neuronales Convolucionales.
- Redes Generativas Adversarias
- Transformadores y Encoders, entre otras.

A continuación, exploraremos en detalle las redes neuronales de perceptrón multicapa, redes neuronales recurrentes, redes neuronales convolucionales y, por último, las redes generativas adversarias.

2.4.2. Redes Neuronales de Perceptrón Multicapa

Las redes de perceptrón multicapa (MLP, por sus siglas en inglés) son uno de los algoritmos pioneros dentro del aprendizaje profundo (AP). Como se mencionó al final de la sección anterior, estas redes derivan de la motivación biológica de recrear sistemas de procesamiento similares al cerebro humano: redes complejas interconectadas de neuronas que, ante cierta información o estímulo, activan las conexiones necesarias para generar una decisión, pensamiento o acción satisfactoria.

El algoritmo de perceptrón multicapa es uno de los tipos de redes neuronales artificiales más estudiados debido a que fue pionero en el campo del AP. La forma en cómo logra realizar su tarea es mediante una combinación lineal de las entradas (variables o características), en donde realiza operaciones matemáticas en cada neurona mediante pesos asignados a cada variable y una función de activación específica. Posteriormente, clasifica a los datos con base en un umbral previamente definido para decidir si pertenece a una categoría u otra.

Según García, R. et al (2020) [28], el perceptrón multicapa se compone de tres partes principales:

- Fase de entrada: Contiene una capa (conjunto de neuronas) que recibe las características u observaciones del conjunto de datos de estudio.
- Fase de procesamiento: Esta fase se compone de una o más capas ocultas en donde transforma los datos y los procesa.
- Fase de salida: Aquí se tiene una capa extra que genera la salida final de la red neuronal, ya sea una regresión o una clasificación.

Cada neurona de las capas antes mencionadas realiza una combinación ponderada de las entradas (sumas ponderadas) más una constante que representa el sesgo del modelo. Posteriormente, aplica una función de activación para transformar los datos. En términos arquitectónicos, el flujo comúnmente es

secuencial: los datos ingresan, se transforman en las neuronas de una capa y la salida de esta capa sirve como entrada para la siguiente etapa, que puede ser otra capa oculta o la capa final que genera el resultado.

La forma en cómo se procesa la información dentro de este algoritmo es mediante el cálculo de valores óptimos de parámetros como pesos, sesgos y funciones de activación. De forma general, dicho proceso sigue estos pasos:

- 1. Primero, se toma un vector de características que sirve como entrada para el perceptrón.
- 2. Cada neurona realiza una combinación ponderada de las características, incorporando pesos y sesgos específicos.
- 3. En seguida, se aplica una función de activación no lineal, la cual permite representar relaciones más abstractas o relaciones no lineales en los datos.
- 4. La salida de cada capa se utiliza como entrada para la siguiente capa, repitiendo el cálculo en n capas hasta llegar a la capa final.
- 5. En la capa de salida, se realiza el cálculo de activación final y se produce una salida, que puede ser una probabilidad (para clasificación) o un valor numérico (para regresión).

Además, el aprendizaje de la red está compuesto por los siguientes pasos:

- 1. Cálculo del error: Aquí se determina la diferencia entre el valor predicho por la red y el valor real, utilizando una *función de pérdida*, como el error cuadrático medio (también conocido como MSE, por sus siglas en inglés).
- 2. Cálculo del gradiente: Para el cálculo del gradiente se utiliza la regla de la cadena, la cual se refiere a la derivada de la función de pérdida respecto a los parámetros de la red, avanzando desde la última capa hacia la primera.
- 3. Ajustes: Por último, se sigue un proceso iterativo que optimice la función de pérdida, ajustando los pesos y los sesgos del modelo para minimizar el error.

Algunas ventajas del perceptrón multicapa son: su capacidad para modelar relaciones no lineales complejas, su flexibilidad y adaptabilidad a distintos tipos de problemas, como clasificación, regresión o series temporales y, su escalabilidad, debido a que puede incluir múltiples capas ocultas para incrementar su capacidad de representación. No obstante, su principal desventaja es la pérdida de interpretabilidad, ya que pequeños cambios en la arquitectura pueden generar modificaciones significativas en los resultados. Además, requiere grandes cantidades de datos y mayor tiempo de entrenamiento. A pesar de estas limitaciones, ha demostrado ser efectivo en diversas áreas como el diagnóstico médico, clasificación de texto, predicción en problemas con series temporales y reconocimiento de voz e imágenes.

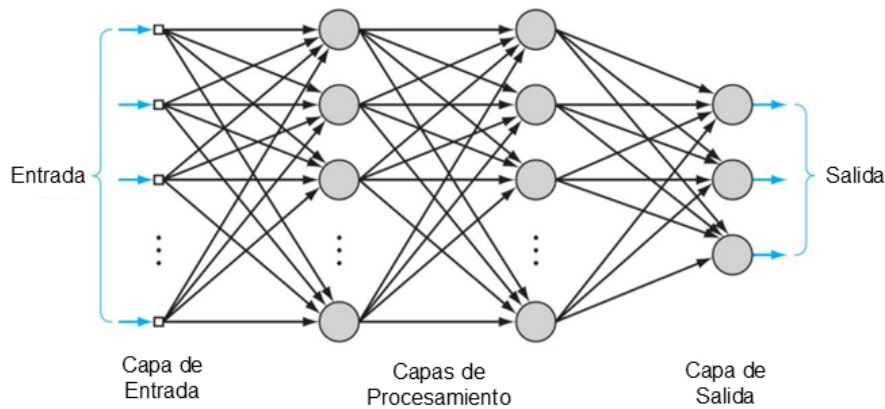


Figura 2.2: Estructura Genérica de una Red de Perceptron Multicapa.
Arana, C (2021) [29]

2.4.3. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNR) constituyen otro tipo de arquitectura cuyo principal diferenciador radica en su capacidad de mantener un *estado interno* o memoria, lo que les permite procesar información de manera secuencial, considerando el contexto y el orden de los datos. Esto las hace especialmente útiles en tareas donde el orden y el contexto son esenciales, como en el procesamiento de lenguaje natural, donde el significado de las palabras depende del contexto [2].

De acuerdo con Rebala, G. et al [23], las RNR se pueden describir como redes de neuronas que procesan información secuencialmente, utilizando los estados anteriores como insumos para el procesamiento de nuevas entradas. En otras palabras, las RNR almacenan información del pasado en un estado oculto que se integra con la entrada actual, creando un aprendizaje basado en contexto temporal.

La forma en cómo funciona las RNR se lleva a cabo en los siguientes pasos:

- 1. Primero llegan los datos como entrada para la red.
- 2. En el primer paso ($t = 1$), se utiliza la entrada inicial para calcular el estado oculto inicial o cero (h_0).
- 3. En cada paso subsiguiente ($t > 1$), se usan las entradas actuales junto con el estado oculto anterior para calcular un nuevo estado oculto que se continuará utilizando en el siguiente paso.
- 4. Cuando se termina la secuencia, se produce una salida que puede ser utilizada ya sea como predicción o para la generación de nueva información.
- 5. La forma en cómo mejora el modelo es mediante ajustes en los parámetros mediante retropropagación de errores.

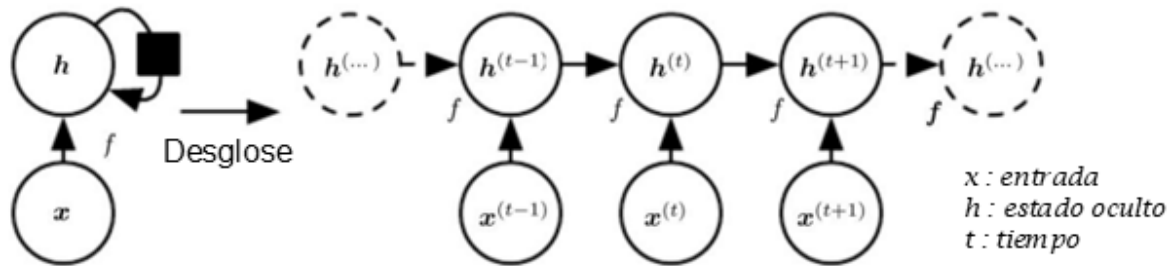


Figura 2.3: Estructura Genérica de una Red Neuronal Recurrente.
Arana, C (2021) [29]

Las principales ventajas de las RNR son su capacidad para modelar dependencias temporales o contextuales y generar secuencias estructuradas, y su amplia aplicación en problemas como análisis de texto, procesamiento de audio y series temporales. Sin embargo, presentan desventajas importantes como la dificultad de entrenar con secuencias largas, tiempos de procesamiento más altos y una limitada capacidad de paralelización, debido a su naturaleza secuencial, en comparación con otras arquitecturas.

2.4.4. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (RNC) se desarrollan como una técnica para analizar información contenida en imágenes, surgida de la necesidad de reconocer información gráfica en nuestro entorno cotidiano [30]. Al igual que nuestro cerebro, que interpreta patrones, volúmenes y texturas, las CNN son capaces de identificar patrones comunes (como una cara en el tronco de un árbol) debido a esos patrones repetidos.

La base de conocimiento de una red neuronal convolucional consiste en reconocer patrones visuales a través de imágenes. La unidad mínima de una imagen se conoce como *píxel*, cuyo valor representa información. Dicha información puede corresponder tanto a representaciones en blanco y negro como a color. Por ejemplo, en una representación en blanco y negro, el valor del píxel podría variar entre 0 y 255. Estos valores son fundamentales para identificar patrones, ya que cada uno influye en la interpretación de los valores adyacentes para que, en conjunto, formen la composición visual completa.

Consistentemente, las CNNs aplican múltiples capas de filtros, usando la operación matemática de convolución. Estos filtros, o kernels, generalmente se representan como matrices que transforman los píxeles originales. Algunos parámetros importantes en las CNNs incluyen el tamaño de los filtros, que determina cuántos píxeles se modifican con cada operación de convolución. Por ejemplo, un filtro de tamaño 3x3 impactará 9 píxeles en la imagen.

Cada operación de convolución se obtiene de pasar los valores de los píxeles a través de los filtros, esto

genera un *mapa de características* que logra resaltar los patrones relevantes que la red busca identificar. Este es un proceso que se vuelve iterativo para todos los píxeles de la imagen y, si se trata de una imagen a color, se multiplica por el número de canales RGB, es decir, se multiplicaría por 3.

Los datos resultantes de la convolución son procesados por capas posteriores. Como indicó Zhao et al. (2023) [31], una configuración típica de CNN incluye capas de convolución para extraer mapas de características; capas de activación para modificar la información según funciones no lineales; y capas de submuestreo (pooling) para reducir la dimensionalidad y evitar el sobreajuste. Este ciclo se repite en diferentes órdenes y dimensiones, según la arquitectura de la red, culminando en una capa completamente conectada que adapta la información a un vector unidimensional, facilitando la predicción o clasificación.

El objetivo de aplicar filtros y convoluciones es simplificar progresivamente la dimensionalidad mientras se compacta la información, permitiendo identificar patrones más complejos en etapas posteriores. Si anteriormente 9 píxeles se representaban en una matriz de 3x3, al aplicar convolución, la información se comprime a un único valor, conservando las características esenciales para la identificación de patrones.

Una RNC aprende mediante el ajuste de sus filtros aplicando el algoritmo de retropropagación. En síntesis, este proceso conlleva lo siguiente:

- 1. Propagación hacia adelante: La imagen de entrada recorre las distintas capas, recibiendo convoluciones y transformaciones sucesivas, concluyendo en una capa que clasifica o predice.
- 2. Cálculo del error: Se determina la discrepancia entre el valor predicho por la red y el valor real, mediante una *función de pérdida*, como el promedio del error cuadrático (MSE por sus siglas en inglés).
- 3. Cálculo del gradiente: Aquí, tal y como lo vimos en otras redes, se utiliza la regla de la cadena, en donde se deriva la función de pérdida respecto a cada parámetro de la red, siguiendo un sentido de adelante hacia atrás.
- 4. Actualización de parámetros: Finalmente se ajustan los parámetros de la red para minimizar el error de forma iterativa hasta minimizar el error.

2.4.5. Redes Neuronales Generativas Adversarias

Habiendo abordado las redes neuronales convolucionales, procederemos ahora a examinar en este documento las Redes Neuronales Generativas Adversarias (GAN, por sus siglas en inglés), que surgen de una disciplina en pleno auge al momento de redactar esta tesis. La Inteligencia Artificial Generativa (GenAI) tiene un impacto significativo en varios campos de la creatividad y el conocimiento humano. Si bien este tema no es el foco principal de nuestro estudio, se le dará una breve consideración dado su pertinencia.

Esta nueva rama representa en la actualidad un avance muy significativo en términos de eficiencia en tiempo y generación de contenido. Uno de sus objetivos muy polémicos es la automatización de tareas asociadas con la creatividad humana en la que se requieren de procesos cognitivos complejos y tardados.

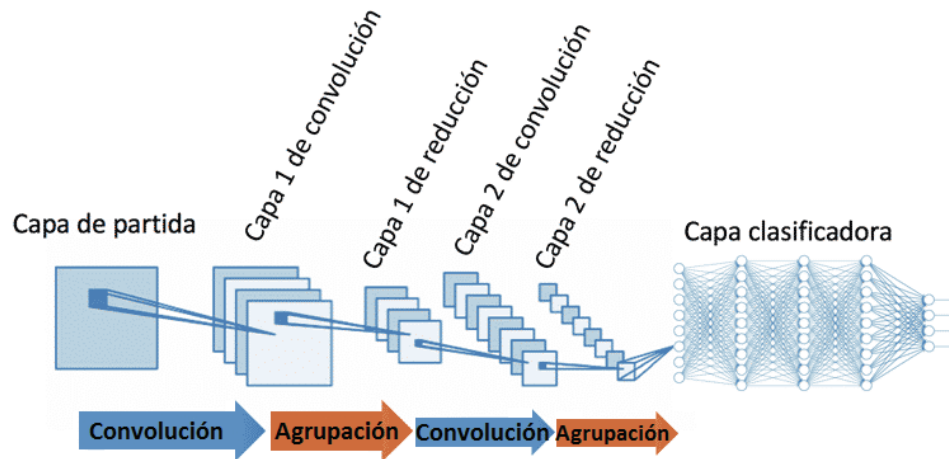


Figura 2.4: Estructura Genérica de una Red Neuronal Convolutional.
Imagen retomada y traducida de Calvo, P. [32]

Algunos de los campos en los que esta área es particularmente utilizada son la generación de texto, composición de música, diseño de imágenes y generación de códigos de programación.

Las Redes Neuronales Generativas Adversarias (GAN) demuestran ser eficaces en la generación artificial de contenido, particularmente en imágenes y videos, así como en la generación de datos sintéticos en disciplinas como medicina, química, meteorología y geología [30].

El concepto de Red Generativa Adversaria se refiere al mecanismo por el cual opera esta estructura de red: se compone de dos redes neuronales principales, el *generador* y el *discriminador*. Según lo describe De Souza et al. (2023) [30], estas redes participan en un proceso competitivo; el generador intenta crear contenido sintético basado en datos de prueba, mientras que el discriminador evalúa y diferencia el contenido real del generado. El objetivo es optimizar la capacidad del generador de producir contenido indistinguible del real para el discriminador.

Los autores explican que la red generadora produce una salida sintética empleando datos reales de prueba, la cual el discriminador evalúa y clasifica como real (con un valor de 1) o sintética (con un valor de 0). Ambos componentes ajustan sus parámetros mediante retropropagación, como se discutió anteriormente. Este proceso iterativo entre las dos redes se optimiza minimizando una función de costo, buscando alcanzar el mínimo error posible.

Una vez generados los datos sintéticos, el siguiente paso consiste en verificar si estos realmente muestran similitudes con los datos reales. Existen varias metodologías para evaluar la similitud entre ambos conjuntos de datos, pero de manera general, el objetivo principal es analizar la distribución de los datos sintéticos y comprobar, a través de una inspección visual inicial, si están adoptando patrones similares a los de la información original.

Además de la evaluación visual, es posible emplear técnicas más específicas para validar la similitud, como el cálculo de la raíz del error cuadrático medio (*Root Mean Square Error*, RMSE, por sus siglas en Inglés) o pruebas estadísticas de distribución, como la prueba de Mann-Whitney U. Aunque este último enfoque no será abordado en detalle ya que no forma parte de los objetivos principales de esta tesis, se

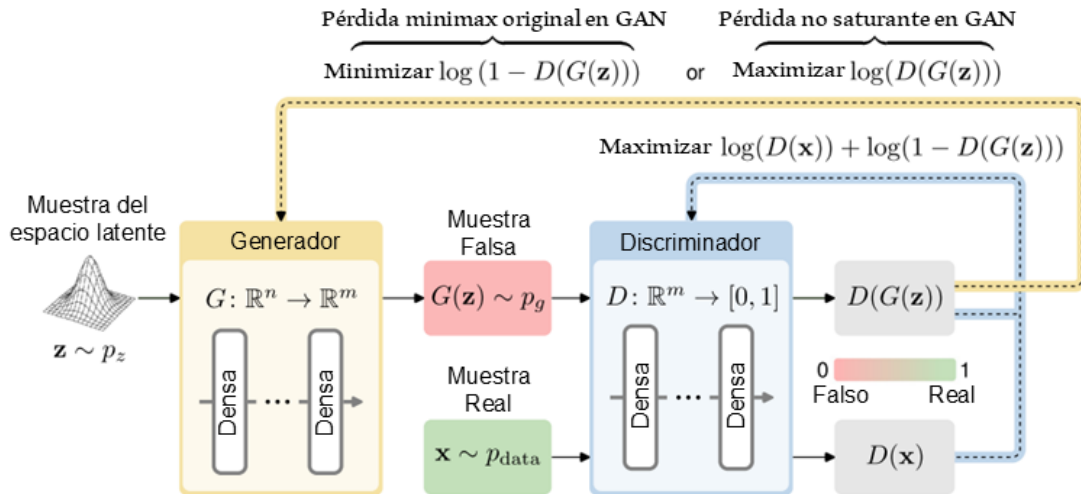


Figura 2.5: Estructura Genérica de una Red Neuronal Generativa Adversaria.
De Souza et al. (2023) [30]

menciona como una herramienta adicional que podría aplicarse en otros contextos.

El proceso completo de generación de datos sintéticos mediante redes neuronales generativas adversarias (GAN) involucra varias etapas iterativas. Inicialmente, los datos sintéticos son evaluados por la parte discriminante de la GAN, y los resultados son analizados para identificar áreas de mejora. Este procedimiento se repite hasta que se obtengan candidatos viables de datos sintéticos con un comportamiento y patrones consistentemente similares a los datos reales. Finalmente, se realizan comparaciones, pruebas estadísticas y análisis matemáticos para comprobar que los datos generados se asemejan fielmente a los originales. Una vez completado este proceso, se puede considerar que la etapa de generación de datos sintéticos ha concluido satisfactoriamente.

En la siguiente sección, se aborda la extracción de características dentro de la Inteligencia Artificial. Aquí se resaltan dos técnicas: el análisis de componentes principales (ACP) y el análisis discriminante lineal (ADL).

Sección 2.5.

Extracción de Características

En el estudio de la Inteligencia Artificial, existe un paradigma que sugiere que un mayor volumen de información no siempre facilita la resolución de un problema, sino que puede complicarlo. Este fenómeno, conocido como la Maldición de la Dimensionalidad, destaca que aumentar las variables en un modelo de Aprendizaje Automatizado puede, en lugar de ser beneficioso, sobrecargar el modelo con información irrelevante, lo que degrada su rendimiento. Por lo tanto, es crucial contar con métodos que nos permitan identificar las variables más relevantes para nuestra tarea, evitando así la inclusión de datos superfluos. En esta sección, se explora la importancia de la Maldición de la Dimensionalidad y se discute los dos métodos de extracción de características que fueron aplicados durante el desarrollo de este trabajo.

Comenzamos analizando la *Maldición de la Dimensionalidad*, un fenómeno que surge de considerar todas las variables posibles de un sistema. Este término se refiere al aumento exponencial de la dimensionalidad de los datos. Para comprender mejor este concepto, consideremos el siguiente ejemplo:

Podemos representar cada variable como una dimensión en el espacio. Si tratamos de abordar un problema con una sola variable, encontraremos dificultades; sin embargo, al añadir una segunda dimensión, obtenemos un plano que facilita la comprensión. Al incorporar una tercera dimensión, podemos imaginar el universo con longitud, latitud y profundidad. Sin embargo, agregar más dimensiones, como la cuarta, quinta o n dimensiones adicionales, hace que la visualización se torne compleja. Aunque no estamos modelando el universo, esta analogía ilustra cómo aumenta la complejidad en los modelos al intentar considerar todas las variables posibles en lugar de centrarse en las que realmente son significativas.

Para evitar caer en la maldición de la dimensionalidad, es crucial que la correlación existente entre las variables seleccionadas sea baja. Pues si no se maneja así, las variables podrían considerarse como redundantes para el modelo y no lograrían aportar información relevante para el objeto de estudio. Así pues, este proceso busca identificar las variables más significativas mediante métodos específicos.

Uno de estos métodos es el Análisis de Componentes Principales (PCA, por sus siglas en inglés), un procedimiento estadístico que permite identificar los elementos más significativos para representar de manera aproximada, pero efectiva, la información general de los datos. Este método simplifica la complejidad de espacios muestrales de muchas dimensiones, conservando la información más representativa [33].

Para entender mejor la teoría detrás de este método, debemos definir los eigenvectores y eigenvalores. En términos sencillos, los eigenvectores, o vectores propios, son aquellos que conservan su dirección al multiplicarse por un escalar (eigenvalor o valor propio). Estos vectores son fundamentales para obtener los componentes principales, ya que, al ser ortogonales entre sí, aseguran la ausencia de correlaciones, mientras que los eigenvalores indican la varianza asociada a cada vector. En síntesis, los componentes principales (que en álgebra lineal se conocen como *eigenvectores*, son combinaciones lineales independientes que retienen la máxima cantidad de varianza (dispersión de los datos con respecto a la media), y los cuales mejoran la representación de los datos con menos variables.

El proceso para calcular los componentes principales de un conjunto de datos incluye la estandarización de los datos, el cálculo de su matriz de covarianza, la obtención de los eigenvectores y eigenvalores a partir de esta matriz, establecer un umbral de varianza acumulada para decidir el número de componentes principales, examinar la relevancia de cada variable respecto a estos nuevos componentes, y eliminar aquellas que no cumplan con un umbral definido.

Para finalizar con esta técnica, veamos el siguiente ejemplo:

Imagina que estás analizando los gustos de las personas basándote en dos aspectos: gusto por los libros y gusto por las películas. Tienes una lista de personas (conjunto de datos), y quieres reducir estos dos parámetros a uno solo para simplificar el análisis (quieres encontrar un componente principal).

Tus datos son puntos en un plano cartesiano, donde el eje X representa el gusto por los libros y el eje Y el gusto por las películas. Cada persona es un punto en este plano. Entonces, una persona (Juan Perez) puede tener un gusto por los libros en una escala de 0 a 10 un 9 y un gusto por las películas un 4, es decir, Juan Perez (9,4). Un ejemplo de cómo se vería esto con muchas personas sería:

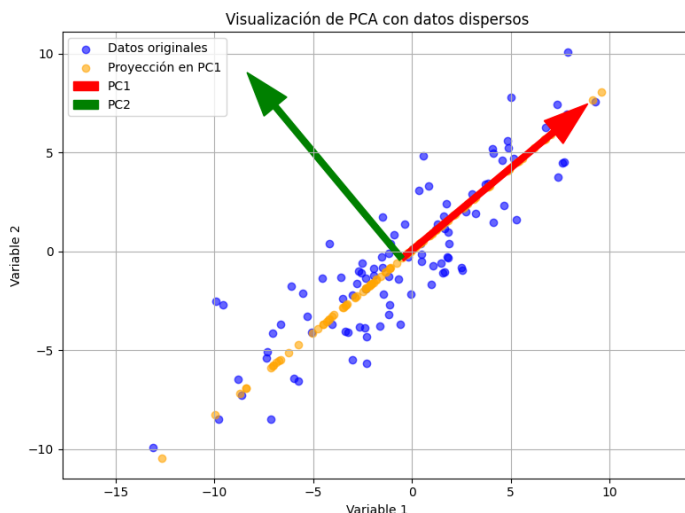


Figura 2.6: Ejemplo ilustrativo de Análisis de Componentes Principales.
Imagen generada con Inteligencia Artificial [34]

El PCA analiza la distribución de todos estos puntos y encuentra una dirección (denominada *componente principal*), que en 2.6 serían las flechas verde y roja. Después se decide cuál componente se elige con base en la mayor cantidad de varianza en los datos (por ejemplo, la flecha roja).

Luego proyecta todos los puntos sobre este nuevo eje (puntos amarillos en la imagen). Por último, esto reduce tus dos variables (libros y películas) a una sola, que puede interpretarse como un “eje general de preferencia cultural” y con esto habrías reducido tu dimensión a una sólo variable.

El Análisis Discriminante Lineal (LDA, por sus siglas en inglés) es una técnica de reducción de dimensionalidad y clasificación que se utiliza para encontrar una combinación lineal de características que separa de manera óptima dos o más clases de datos. A diferencia de otros métodos que buscan maximizar la varianza de los datos, LDA se concentra en maximizar la separación entre las clases.

Esta técnica tiene como objetivo proyectar la información en un espacio de menores dimensiones sin perder la capacidad de discriminar entre las clases del problema. La forma en como lo logra es maximizando la distancia entre las medias de cada clase, mientras que minimiza la varianza dentro cada una de ellas.

Segun lo descrito por Balakrishnama et al (1998) [35] el procedimiento para aplicar LDA comienza con el cálculo de las medias de cada clase y la media global de todas las clases. Posteriormente, se calcula la matriz de dispersión dentro de las clases (intra-clase) y la matriz de dispersión entre las clases (inter-clase). La matriz de dispersión intra-clase representa la varianza dentro de cada clase, mientras que la matriz de dispersión inter-clase refleja la varianza entre las medias de las clases.

Posteriormente, se utilizan estas matrices para determinar una función discriminante lineal para proyectar cada dato en un espacio de menores dimensiones. Esto implica que se encuentren los eigenvectores de la matriz que resulta de la relación entre las matrices de dispersión, puesto que dichos eigenvectores representan las direcciones que maximizan la separación de los datos.

En resumen, la extracción de características representa un proceso fundamental al momento de

trabajar con modelos de AA, pues buscan que el modelo sea lo más general posible para que sea robusto ante nuevas entradas, eficiente en términos computacionales y confiable con resultados explicativos.

Sección 2.6.

Métricas de Evaluación

Durante esta sección se abordan las métricas de evaluación, las cuáles se refieren a medidas esenciales que cuantifican el rendimiento de un modelo de Inteligencia Artificial. Dichas métricas identifican las áreas donde un modelo podría estar fallando, así como también para medir la efectividad de dicho modelo para cumplir con su objetivo. Así pues, se exploran las principales métricas de evaluación utilizadas para la tarea de clasificación, proporcionando sus definiciones y sus fórmulas para obtenerlas. Aunado a esto, también se incluyen ejemplos prácticos de estas medidas para facilitar su interpretación y aplicación de éstas.

Según planteó Hossin, M. et al (2015) [36], existen diversas métricas de evaluación que ayudan a entender cómo se desempeña el modelo en la tarea de clasificación, lo cual permite identificar áreas de mejora y ajustar los algoritmos para lograr resultados más precisos.

A continuación, se enlistan estas métricas:

- **Exactitud (*Accuracy*):** La exactitud es la métrica más básica y se calcula como la proporción de predicciones correctas sobre el total de predicciones realizadas. Aunque es intuitiva, la precisión puede no ser la mejor medida cuando hay un desequilibrio en las clases, ya que no considera la distribución de los datos. En un modelo de clasificación de correo electrónico para detectar spam, si el modelo clasifica correctamente 90 de 100 correos (incluyendo spam y no spam), la precisión es del 90%. Sin embargo, si en realidad hay 95 correos no spam y 5 spam, la precisión podría no reflejar la efectividad en identificar spam específicamente.
- **Precisión (*Precision*):** Esta métrica explica la proporción de verdaderos positivos sobre el total de positivos predichos. Esto significa una visión sobre la calidad de las predicciones positivas en el modelo y es muy útil cuando el impacto de un falso positivo es grave. Por ejemplo, en una aplicación del área de la salud para detectar enfermedades, si de 20 diagnósticos positivos, 15 son pacientes verdaderamente enfermos y 5 son falsos positivos, la precisión es del 75%. Esta métrica es de mucha ayuda, pues es muy grave tratar a un paciente de una enfermedad que no tiene.
- **Sensibilidad (*Recall*):** A esta medida se le conoce también como tasa de verdaderos positivos y representala capacidad del modelo para identificar correctamente los verdaderos positivos. El concepto es similar al de la métrica de exactitud, sin embargo, aquí no toman lugar los falsos positivos, en otras palabras y regresando al ejemplo médico, sólo te interesa si los que clasificaste bien son verdaderos.
- **Medida F1 (*F1 Score*):** Esta medida proporciona una noción del equilibrio entre la precisión y la sensibilidad. Es una métrica muy valiosa cuando las clases del objeto de estudio no están equilibradas (hay más de una clase que de otra). Lo que transmite esta métrica es la capacidad de tu modelo para no tener falsos positivos o falsos negativos. Por ejemplo, para la detección de

fraudes financieros, donde tanto los falsos positivos como los falsos negativos son costosos, un alto F1 Score indica que el modelo maneja adecuadamente ambos tipos de error, pues estarías evitando fraudes.

- **Curva ROC y AUC:** La curva de Característica Operativa del Receptor (ROC, por sus siglas en inglés) representa gráficamente el desempeño del modelo para diferentes umbrales de decisión. El Área Bajo la Curva (AUC) es una métrica que mide el rendimiento del modelo en total, donde un valor cercano a 1 indica un modelo excelente. Por ejemplo, un modelo de predicción de riesgos de crédito con un AUC de 0.9 indica que el modelo tiene alta capacidad de discriminar entre buenos y malos riesgos.
- **Especificidad (*Specificity*):** También conocida como tasa de verdaderos negativos, la especificidad mide la proporción de casos negativos que el modelo identifica correctamente sobre el total de casos negativos reales. Es fundamental cuando el costo de los falsos positivos es alto y se necesita garantizar que las instancias negativas sean clasificadas con precisión. En un sistema de control de acceso por reconocimiento facial, si de 100 personas no autorizadas, el sistema rechaza correctamente a 95 y permite el acceso erróneamente a 5, la especificidad es del 95 %.

Cada una de estas métricas ofrece información valiosa sobre el desempeño del modelo, y su relevancia depende del contexto específico del problema que se está abordando. En campos como la medicina o el fraude financiero, es más relevante maximizar la sensibilidad que la precisión, por ejemplo.

Experimentación

Una vez abordados los temas fundamentales que establecen las bases teóricas para comprender los algoritmos empleados en esta tesis, en este capítulo se describen las acciones implementadas y los procesos que permiten obtener los resultados que se presentan en el siguiente apartado.

Los experimentos presentados en este capítulo se diseñaron con base en los datos provenientes del proyecto colaborativo descrito en la Sección 1.1, cuyas características —un conjunto originalmente reducido, desbalanceado y con restricciones de privacidad que limitan el uso de variables demográficas— motivaron el desarrollo de dos enfoques complementarios: el uso de escalogramas y la generación de datos sintéticos mediante redes neuronales generativas adversarias. La evaluación de todos los modelos implementados se realizó de manera uniforme, empleando las métricas definidas en la Sección 2.6: ROC-AUC, Precisión, Exactitud, Sensibilidad, Especificidad y Medida F1, siendo la curva ROC-AUC, el criterio principal de ordenamiento y comparación entre modelos, dado que cuantifica la capacidad global del modelo para distinguir entre clases independientemente del umbral de decisión, lo cual es especialmente relevante ante conjuntos de datos desbalanceados. La comparación entre modelos se llevó a cabo mediante mapas de calor que permiten visualizar simultáneamente el desempeño de cada configuración a través de todas las métricas consideradas, facilitando la identificación de las configuraciones óptimas en cada escenario experimental.

Antes de detallar los pasos de los experimentos, que se organizan en diferentes secciones según las particularidades de cada parte del proyecto, es relevante abordar los tratamientos preliminares que se realizan sobre los datos antes de la aplicación de los algoritmos seleccionados. Aunque este tratamiento no constituye el objetivo central de la tesis, su descripción resulta esencial para comprender la metodología utilizada en la generación y preparación de la información analizada. Específicamente, se describen los procedimientos realizados con los datos obtenidos de los estudios de electroretinogramas (ERG), tanto para la creación de los escalogramas como para la generación de datos sintéticos mediante redes neuronales generativas adversarias (*Generative Adversarial Networks*, GAN, por sus siglas en inglés).

Los datos en bruto provenientes de los pacientes sometidos a estudios de ERG se procesan para extraer, segmentar y organizar los valores de los potenciales eléctricos obtenidos dentro de un rango específico de frecuencias. Este procesamiento preliminar es fundamental para estructurar de manera adecuada los datos, facilitando su uso en las etapas posteriores del análisis. En trabajos previos realizados en el marco de este proyecto, dicho conjunto de datos se somete a una serie de transformaciones matemáticas, generando un nuevo conjunto de datos que posteriormente sirvió para la construcción de imágenes de escalogramas mediante un segundo código de programación. Estas imágenes se utilizaron como insumos para la implementación de algoritmos de redes neuronales convolucionales (*Convolutional*

Neural Networks, CNN). Para una descripción más detallada acerca de la obtención y tratamiento de este conjunto de datos, se recomienda consultar el artículo de E. Bárcenas et al. [3].

Para la generación de datos sintéticos, los datos pasan por un tratamiento adicional mediante redes neuronales generativas adversarias. Esta fue una decisión que se tomó con base en el objetivo de mejorar los resultados reportados en el artículo mencionado en el párrafo anterior. Este nuevo enfoque también tuvo lugar para evitar los tiempos prolongados que se requieren para realizar nuevos estudios con pacientes reales. Así pues, se implementó una arquitectura específica del cuál no se hablará en esta tesis, para crear nuevos registros con base en los datos reales. Aunado a esto, a estos datos se les aplicaron pruebas matemáticas para validar que su comportamiento y distribución fuera lo más similar posible a los de los pacientes, pues esto asegura resultados confiables y representativos para el objetivo central de este trabajo y en general, del proyecto mencionado en la introducción.

Por último, para una mejor organización y comprensión, se han dividido los experimentos en dos subsecciones, que corresponden a cada tipo de datos utilizados. La primera subsección, denominada Escalogramas, se destina a los experimentos realizados con las imágenes generadas a partir de los estudios de ERG. La segunda subsección, llamada Datos Sintéticos, documenta los experimentos desarrollados bajo la generación automática de datos utilizando redes neuronales generativas adversarias.

En cuanto a los experimentos realizados con los datos sintéticos, se elaboran tres configuraciones distintas basadas en la cantidad de registros generados, empleados junto con los datos reales procesados mediante la transformación matemática Wavelet Morlet real. Las configuraciones experimentales incluyen:

- x1: Se obtiene un conjunto de datos compuesto por los registros reales y un conjunto de datos sintéticos del mismo tamaño que el conjunto real.
- x10: Se tiene un conjunto de datos que incluyó los datos reales junto con información sintética, aumentando el número total de muestras con un factor 10 veces.
- x100: Se realiza con un conjunto que combinó los datos reales con información sintética, incrementando el número total de muestras con un factor 100 veces.

Sección 3.1.

Preprocesamiento

Para el caso de los escalogramas, una vez etiquetadas todas las imágenes con las clases correspondientes, “disorder” y “health” para enfermos y sanos, respectivamente, se procede a extraer las imágenes almacenadas en el directorio principal. Siguiendo este proceso, cada archivo se abre, se guarda en un arreglo, se redimensiona, se normaliza y su correspondiente etiqueta se registra en otra lista, transformándola a valores numéricos: Disorder = 0 y Health = 1. Posteriormente, con fines de monitoreo, se imprimen en pantalla datos como el número total de imágenes, el número de etiquetas generadas, así como la cantidad de observaciones clasificadas como sanos y enfermos.

Posteriormente, los datos se dividen en conjuntos de entrenamiento y de prueba a una razón de 80/20

respectivamente. Aunado a esto, se realiza la configuración para que esta división sea de forma aleatoria para tratar de eliminar algún sesgo que pudiera existir en las clases.

Finalmente, se renombran las variables obtenidas para facilitar su uso durante la aplicación de los modelos. Para las etiquetas de los conjuntos de entrenamiento y prueba se utiliza una función que adapta el formato de los arreglos al requerido por las redes neuronales convolucionales. Las etiquetas se renombran como “y_test_rnc” y “y_train_rnc”, mientras que las características predictoras se denominan “X_test_rnc” y “X_train_rnc”.

En el caso de los datos sintéticos, se sigue un procedimiento similar al descrito en la preparación de los escalogramas. En primer lugar, se realizan muestreos aleatorios para generar 10 conjuntos distintos con el objetivo de minimizar posibles sesgos en los datos y garantizar una distribución equilibrada entre ambos grupos. Cada conjunto es dividido aleatoriamente en un 80 % para entrenamiento y un 20 % para prueba.

El proceso para cada uno de los conjuntos incluye los siguientes pasos:

- Validación del conjunto: Se imprime en la línea de comandos el estado actual del procesamiento, especificando el conjunto que estaba siendo ejecutado. Este procedimiento permite un control durante las interrupciones del sistema, facilitando la reanudación del proceso desde el último conjunto trabajado.
- Validación de dimensiones: Aquí se verifica las dimensiones de los conjuntos de entrenamiento y prueba. Aunque es algo que ya se realizó con anterioridad, se imprime en consola para fines de monitoreo.
- Transformación de las clases objetivo: Las clases objetivo, almacenadas en la variable “health.status”, se transforman de representaciones textuales (“Disorder”, “Health”) a valores numéricos mediante la conversión a un DataFrame, y después se guardan nuevamente en un arreglo con las modificaciones realizadas: Disorder = 0 y Health = 1.
- Selección de variables predictoras: Se elimina la columna “health.status”, junto con otras variables como edad y número de paciente, que no aportan información predictiva directamente aplicable para el modelo.
- Estandarización de datos: Se aplica una estandarización de la información para garantizar que todas las variables tengan una escala igual de representativa para todas. Esta estandarización se aplica tanto para el conjunto de entrenamiento como para el de prueba.
- Renombramiento de variables: Las etiquetas de los conjuntos de prueba y entrenamiento se renombran como “y_test” y “y_train”, al mismo tiempo que las variables predictoras se renombran a “test” y “train”, respectivamente.

Sección 3.2.

Extracción de Características

Para los escalogramas se realizan dos métodos de extracción de características. Primero se aplica el *Análisis de Componentes Principales* (PCA) y, posteriormente el *Análisis Discriminante Lineal* (LDA).

El procedimiento de la primer técnica se describe a continuación:

- Definición de los componentes principales: Aquí se definen 4 componentes principales, los cuales explican aproximadamente el 95 % de la varianza de los datos.
- Proyección de las variables predictoras: Los componentes principales se utilizan para proyectar las variables predictoras de los conjuntos de entrenamiento y prueba en dichos componentes.
- Renombramiento de las variables: Las proyecciones obtenidas se almacenan en arreglos y se renombran como “X” para poder hacer la división de los datos como lo que se explica en la fase de preprocesamiento.

Para el caso del análisis discriminante lineal (ADL), los pasos son los siguientes:

- Aplanamiento de las imágenes: Primero se transforman las imágenes en vectores unidimensionales para ser procesados correctamente por el modelo del ADL.
- División en conjuntos de entrenamiento y prueba: Se asegura que los conjuntos de entrenamiento y prueba guarden una proporción de 80 % y 20 %, respectivamente. Al igual que explicaciones anteriores, se establece que esta división sea aleatoria.
- Aplicación de la transformación: Aquí se utiliza la técnica del análisis discriminante lineal para reducir el número de variables a un solo componente que guarda la mayor cantidad de información relevante para la clasificación de los pacientes.

En el caso de los datos sintéticos se aplica el *Análisis de Componentes Principales* (ACP) como método de extracción de características. Los pasos realizados son los siguientes::

- Definición de los componentes principales: Se definen 3 componentes principales que explican aproximadamente el 85 % de la varianza de los datos.
- Proyección de las variables predictoras: Los componentes principales son utilizados para proyectar las variables predictoras de los conjuntos de entrenamiento y prueba en estos componentes.
- Renombramiento de las variables: Las proyecciones obtenidas se guardan con los nombres “X_train” y “X_test”, correspondientes a los conjuntos de entrenamiento y prueba, respectivamente.

Sección 3.3.

Implementación de Modelos y Optimización de Hiperparámetros

Para los datos obtenidos con escalogramas, se realizan diversas modificaciones a los arreglos que contienen las variables predictoras y etiquetas, asegurando que sus dimensiones sean las correctas para ser introducidas en los modelos. La optimización de hiperparámetros se lleva a cabo evaluando los siguientes parámetros:

- Filtros: Se prueban valores de 1 a 64.
- Tamaño del Kernel: Se prueban valores de 1 a 4.
- Función de Activación: Se prueban algunos como 'sigmoid', 'softmax', 'softplus', 'softsign', 'tanh', 'selu', 'elu', 'exponential' y 'relu'.
- Paso del Kernel: Se prueban valores de 1 a 4.
- Taza de Aprendizaje; Se prueban valores entre 0.0001, 0.0004.
- Intentos: Donde se prueban valores como 10, 100, 150 y 200
- Épocas: Donde se prueban valores como 400, 500, 600, 700 y 1000.

En el caso de los datos sintéticos, se optimizan los hiperparámetros en el primer conjunto, denominado conjunto 0, con el fin de determinar los mejores valores para cada parámetro. Estos valores posteriormente se utilizan para entrenar los modelos en los 9 conjuntos restantes, lo que disminuye los tiempos de cómputo requeridos para la experimentación.

Los algoritmos que se utilizan y los hiperparámetros que se optimizan son los siguientes:

- Regresión Logística: Inversa de la fuerza de regularización (C), Tolerancia para criterio de parada, Constante añadida a la función de decisión y Algoritmo de optimización.
- Bayes Ingenuo Gaussiano: Suavizado de varianza.
- Máquina de Soporte Vectorial: Kernel, C (parámetro de regularización), Gamma (coeficiente del kernel) y Grado de la función polinomial del kernel.
- Árbol de Decisión: Profundidad del árbol, Mínimo de muestras para dividir un nodo, Mínimo de muestras para dividir una hoja y Estrategia para elegir la división en cada nodo.
- Bosque Aleatorio: No. de estimadores (árboles), Profundidad máxima del árbol, Máximo número de muestras, Mínimo de muestras para dividir un nodo y Mínimo de muestras para ser una hoja.
- Redes de Perceptrón Multicapa: Número de neuronas por cada capa oculta (HL), Capas ocultas (HL), Función de activación, Algoritmo de optimización, Regularización L2 u y Máximo número de iteraciones.

- Redes Neuronales Recurrentes: Unidades, Función de activación, Algoritmo de optimización y Tasa de aprendizaje.
- Redes Neuronales Convolucionales: Número de filtros, Tamaño del kernel, Paso del kernel, Función de activación, Algoritmo de optimización, Tasa de aprendizaje y Número de épocas.

Sección 3.4.

Almacenamiento de Resultados

Para concluir este capítulo, se describen los procedimientos utilizados para almacenar los resultados de la optimización de hiperparámetros y las métricas de evaluación de los modelos generados. Es importante mencionar que la metodología empleada para guardar esta información fue idéntica tanto para los datos derivados de los escalogramas como para los datos sintéticos.

Los resultados se almacenan en Neptune, una plataforma web diseñada para estructurar y registrar información obtenida durante los experimentos de aprendizaje automatizado. El proceso de optimización de hiperparámetros se ejecuta empleando una biblioteca que permite la integración con esta plataforma. De esta manera, los hiperparámetros óptimos, junto con las métricas de evaluación de los modelos, se registran automáticamente en Neptune.

Una vez que la ejecución de todos los experimentos termina, los datos se exportan en formato `csv` para facilitar su consumo. Este archivo se procesa después con bibliotecas de python para generar los mapas de calor y los análisis que se presentan en el próximo capítulo.

Resultados

En este capítulo se presentan los resultados obtenidos de los experimentos finales realizados con los datos derivados de los Escalogramas y los Datos Sintéticos. Los resultados se muestran mediante mapas de calor que, con base en una escala de 0 a 1, indican el rendimiento de los algoritmos según las métricas de evaluación consideradas. Un resultado más cercano a 1 refleja un mejor desempeño del modelo, mientras que valores más cercanos a 0 denotan un rendimiento bajo.

Cada mapa de calor incluye un título en la parte superior que permite identificar el experimento correspondiente, las condiciones bajo las cuales se realiza y si se aplica alguna técnica de extracción de características. En el caso de los Escalogramas, la forma en la que se identifican las condiciones varía ligeramente respecto a los Datos Sintéticos. Estas características específicas de identificación se explican más detalladamente en la descripción que acompaña cada imagen de resultados.

Adicionalmente, cada fila en las matrices del mapa de calor representa un experimento distinto. Las diferencias entre los experimentos pueden involucrar cambios en los algoritmos empleados o modificaciones en los hiperparámetros configurados, como es común en el caso de los Escalogramas.

Las métricas que se utilizan para evaluar el rendimiento de los modelos incluyen los siguientes indicadores: Precisión, Exactitud, Sensibilidad, Especificidad, F1 Score y Curva ROC / AUC.

Sección 4.1.

Descripción

Con el fin de facilitar la comprensión del sistema desarrollado, la Figura 4.1 presenta un diagrama general de la arquitectura implementada. Las señales de entrada corresponden a las frecuencias extraídas de los estudios de electroretinograma (ERG), representadas como un vector de características F_1, F_2, \dots, F_n por paciente. Estas características son procesadas por el conjunto de modelos implementados, los cuales se distinguen según el enfoque utilizado: los algoritmos enmarcados en amarillo corresponden a los experimentos realizados con datos sintéticos, mientras que el enmarcado en azul —Redes Neuronales Convolucionales— corresponde al enfoque basado en escalogramas. La salida del sistema es un valor de diagnóstico binario: 1 para pacientes clasificados como sanos y 0 para pacientes clasificados con principios de retinopatía diabética.

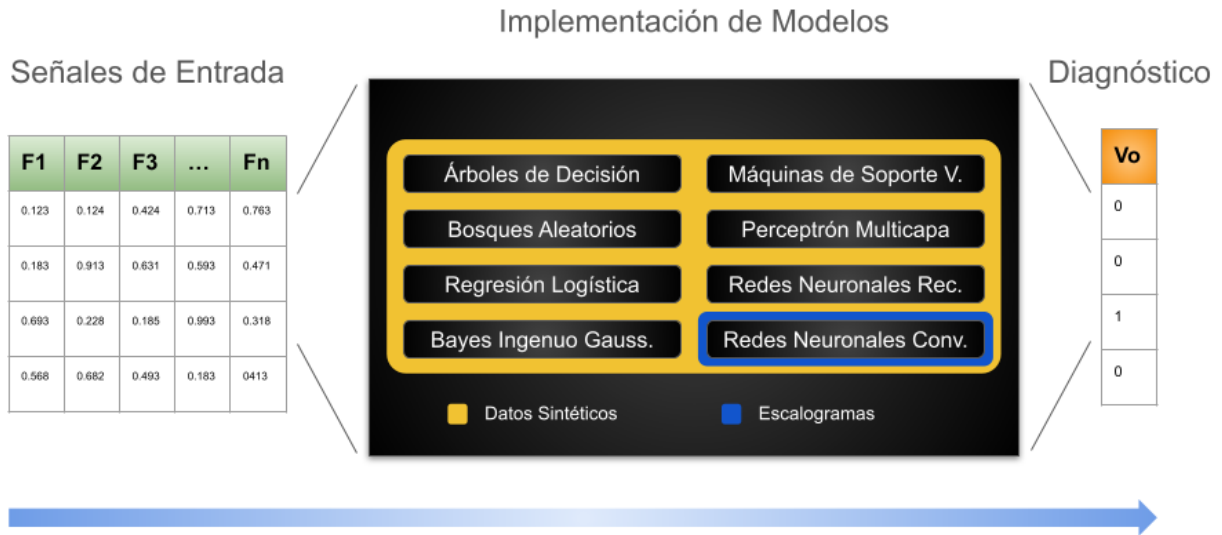


Figura 4.1: Diagrama General del Sistema de Clasificación Implementado.
Imagen de autoría propia.

4.1.1. Escalogramas

Como se muestra en la Figura 4.2, se observa el comportamiento descrito en la introducción de este capítulo. En este caso, los resultados corresponden al entrenamiento del modelo de Redes Neuronales Convolucionales donde se utilizan diferentes configuraciones de hiperparámetros y métodos de extracción de características.

Los nombres asignados a cada fila del mapa de calor representan el tratamiento aplicado en cada experimento y siguen la estructura descrita a continuación:

CNNALL - (parámetro agregado para denotar si se utiliza extracción de características) - #e (número de épocas) - #t (número de intentos) - 2 (para indicar el número del experimento) - #LR (taza de aprendizaje del modelo)

La presentación de este mapa de calor está organizada para que las filas aparezcan ordenadas de acuerdo con el desempeño en la métrica de evaluación *ROC_AUC*. Por convenciones acordadas durante el desarrollo de los experimentos, el experimento con el mejor desempeño bajo esta métrica se ubica en la parte superior de la matriz. Este enfoque permite facilitar la interpretación de los resultados y destacar de forma clara cuál es la configuración óptima generada durante el proceso de experimentación.

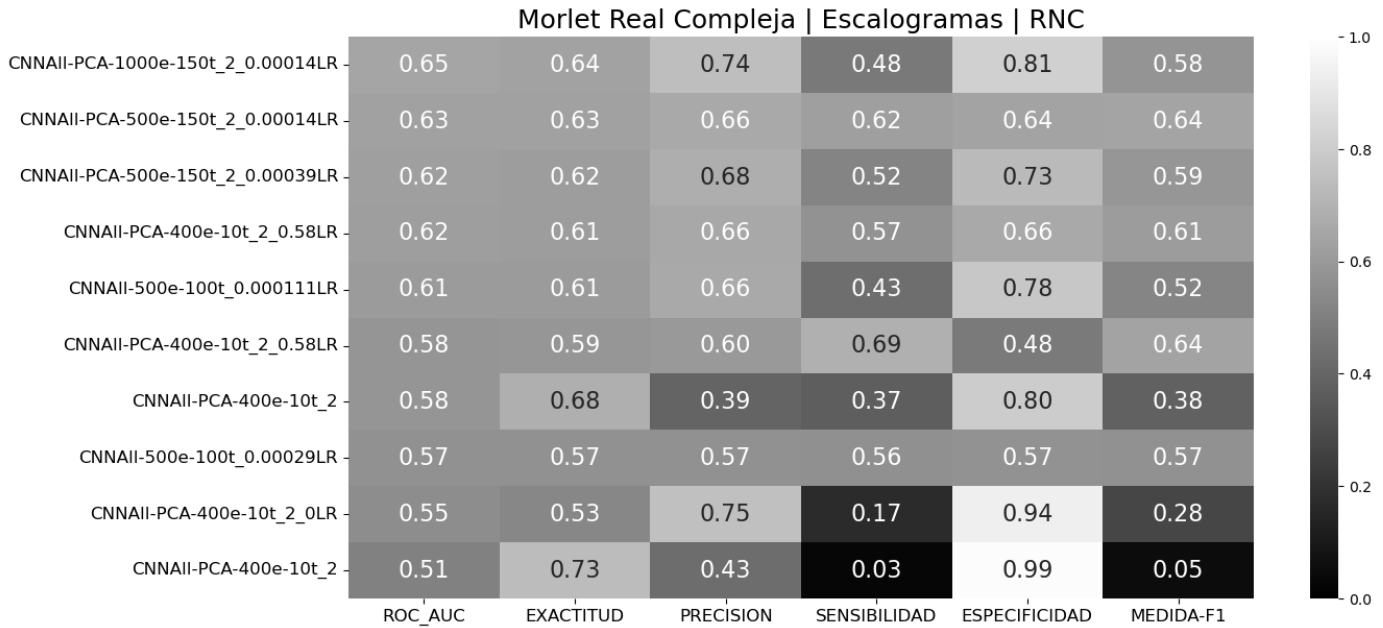


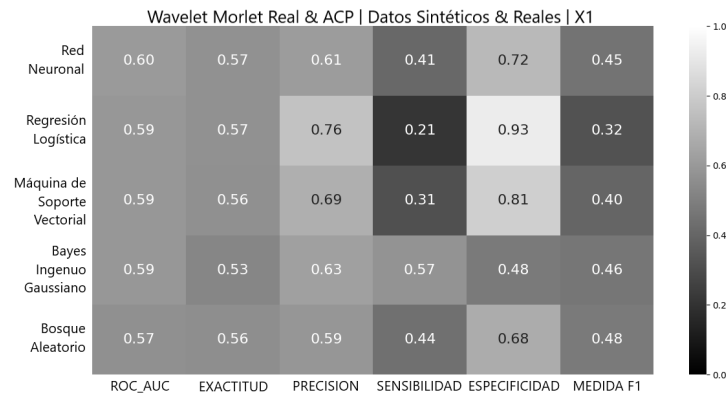
Figura 4.2: Resultados Obtenidos de los Experimentos de Escalogramas.
Imagen de autoría propia.

4.1.2. Datos Sintéticos

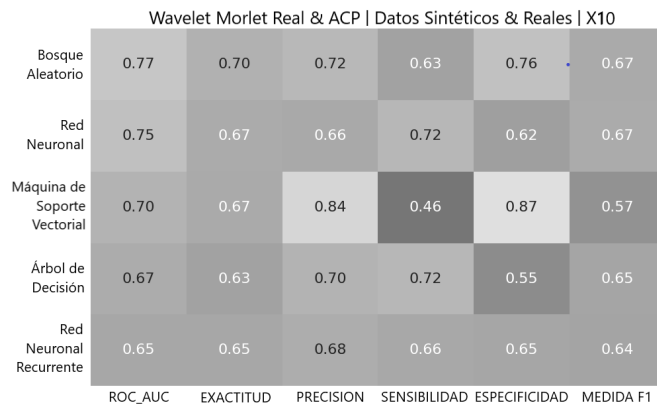
Como se muestra en la Figura 4.3, los resultados corresponden al entrenamiento de los mejores modelos obtenidos para los casos con los datos sintéticos y reales sin aumentar (véase 4.3a), más los resultados con los datos aumentados 10 veces (4.3b) y los resultados con los datos aumentados 100 veces (4.3c).

La forma en cómo se obtienen estos resultados surge de promediar, para cada algoritmo de aprendizaje automatizado, los 10 conjuntos. Es decir, dado que a los 10 conjuntos se les aplica un algoritmo con sus hiperparámetros optimizados, se promedian los resultados para obtener la media del comportamiento de dicho algoritmo. Cada uno de estos mapas de calor cuenta con un título descriptivo que nos dice la transformación matemática que le fueron aplicados a los datos originales, la extracción de características que se utiliza, el tipo de experimento, seguido del indicativo de las veces de datos aumentados (x1, x10 y x100)

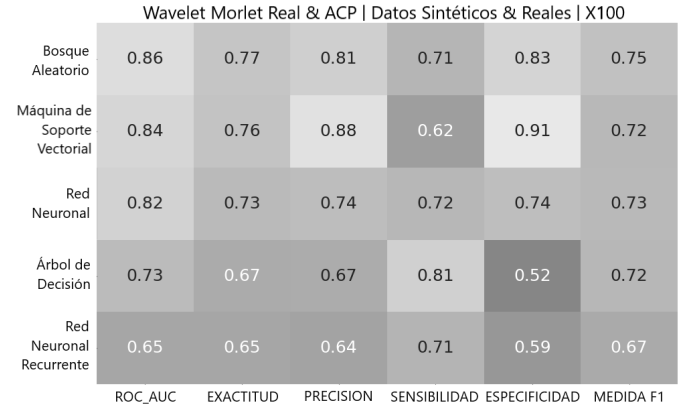
Siguiendo el mismo criterio de ordenamiento descrito para los escalogramas, las filas de cada mapa de calor se organizan de mayor a menor desempeño en la métrica *ROC_AUC*. En este caso, dado que cada renglón corresponde a un algoritmo distinto, se presentan únicamente los 5 modelos con mejor desempeño para cada conjunto de datos (X1, X10 y X100).



(a) Resultados Obtenidos de los experimentos con un factor de X1



(b) Resultados Obtenidos de los experimentos con un factor de X10



(c) Resultados Obtenidos de los experimentos con un factor de X100

Figura 4.3: Resultados Obtenidos de los Experimentos con Datos Sintéticos. Imágenes de autoría propia.

Sección 4.2.

Análisis

Antes de analizar los mapas de calor presentados en la sección anterior, es fundamental hacer un recorrido detallado sobre el proceso que se lleva a cabo para obtener los resultados estudiados. Dicho recorrido comienza con los experimentos realizados en los escalogramas, en los cuales se aplica el algoritmo de Redes Neuronales Convolucionales (CNN, por sus siglas en inglés). Este enfoque es seleccionado debido a los resultados positivos obtenidos en diversas aplicaciones, como el reconocimiento facial en ciberseguridad, la detección de objetos en visión computacional y la identificación de patrones asociados a enfermedades en el ámbito de la medicina. En este contexto, los escalogramas se consideran adecuados para representar, en forma de imágenes tridimensionales (anchura, altura y color), los resultados de las frecuencias seleccionadas a partir de los estudios de electroretinogramas de los pacientes.

El mapa de calor presentado en la Figura 4.2 refleja los primeros resultados obtenidos. En este caso, el experimento con peor desempeño es el entrenamiento del algoritmo RNC bajo configuraciones de 400 épocas y 10 iteraciones. Este experimento tiene como métrica principal de la curva ROC-AUC un

rendimiento del 51 %, lo que implica que el modelo se comporta de manera similar a lanzar una moneda al aire, ya que posee una probabilidad de predicción correcta del 50 %. Aunque este modelo tiene una exactitud de 73 %, lo cual típicamente indica que el 73 % de las observaciones son clasificadas correctamente, otras métricas revelan un desempeño desfavorable. Por ejemplo, la exactitud fue del 43 %, lo que indica que solo el 43 % de las observaciones clasificadas como enfermos realmente pertenecen a esa clase. Asimismo, la sensibilidad muestra un valor crítico del 3 %, reflejando la incapacidad del modelo para identificar correctamente a los pacientes enfermos, mientras que la especificidad es significativamente alta (99 %) y la medida F1 alcanza un valor de apenas el 5 %. De manera general, aunque métricas como la precisión y la especificidad sugieren un rendimiento aceptable en algunos aspectos, el análisis en conjunto pone en evidencia que este modelo resulta ineficaz para clasificar nuevos casos médicos, debido a su incapacidad de identificar correctamente a los pacientes enfermos.

En contraste, el resultado de los escalogramas que alcanza las mejores métricas de evaluación pertenece al algoritmo de RNC en combinación con Análisis de Componentes Principales (ACP), el cual utiliza una configuración de 1000 épocas, 150 iteraciones y una tasa de aprendizaje de 0.00014. Este caso logra una métrica ROC-AUC del 65 %, lo cual significa un desempeño bastante mejor en comparación con el peor resultado que se mencionó en el párrafo anterior. Esto expone que el modelo avanza desde un comportamiento aleatorio hacia un comportamiento donde muestra mejor capacidad para discernir entre pacientes enfermos y sanos. A pesar de que la exactitud (que logra un 64 %) y la especificidad (que alcanza un 81 %) son menores en comparación con el experimento de menor rendimiento, otras métricas como la sensibilidad, el puntaje F1 y la precisión presentan mejores valores que, en conjunto, confirma esta mejora en la capacidad para clasificar nuevos pacientes. Sin embargo, a pesar de presentar mejores resultados, en este ámbito del área de la salud no es nada recomendable contar con un margen de error tan alto, pues es necesario obtener resultados precisos y confiables para diagnosticar un paciente con RD.

Debido a las limitaciones observadas con los escalogramas, se decidió probar un enfoque alternativo basado en la generación de datos sintéticos con el objetivo de mejorar el rendimiento de los experimentos. Como se describió en la sección de definición del problema, el conjunto de datos original presenta un desbalance entre clases. En lugar de eliminar registros de pacientes enfermos para igualar las clases, se optó por generar datos sintéticos adicionales de pacientes sanos para equilibrar la distribución, y posteriormente se realizaron aumentos proporcionales en ambas clases, permitiendo así aprovechar la totalidad de la información disponible.

El primer mapa de calor generado con datos balanceados mediante la transformada Wavelet Morlet Real (4.3a) muestra que el mejor modelo es una Red Neuronal Tradicional (Perceptrón Multicapa). En este experimento, también se evalúan las Redes Neuronales Convolucionales (RNC) utilizando los datos sintéticos, pero a diferencia de los escalogramas, estos datos son transformados solo a un espacio de dos dimensiones para ser procesados como imágenes, sin agregar información adicional como ocurre con los escalogramas. Este aspecto resalta la importancia de aplicar técnicas como la transformada de escalogramas para modelos como RNC.

Aunque el mejor desempeño en el primer mapa de calor corresponde al Perceptrón Multicapa, con mejores resultados frente a algoritmos más estadísticos como la Regresión Logística o el Bayes Ingenuo Gaussiano, y frente a modelos más complejos como las Máquinas de Soporte Vectorial o los Bosques Aleatorios, sus métricas aún son inferiores al mejor modelo basado en escalogramas, lo que refuerza la efectividad de las técnicas se aplican en este último.

El segundo mapa de calor generado con el aumento de los datos en un factor de 10 (4.3b) muestra mejoras considerables respecto a las métricas presentadas en el primer mapa (4.3a). Continuando con esta comparación, los resultados del mejor modelo del primer mapa se ven superados por el peor modelo del segundo, dicho modelo pertenece al algoritmo de Red Neuronal Recurrente que logra un incremento en todas las métricas del 5 % aproximadamente con respecto al mejor resultado del primer mapa (Perceptrón Multicapa).

Ahora bien, el mejor resultado del segundo mapa de calor se obtiene mediante el algoritmo de Bosques Aleatorios, pues alcanza una métrica ROC-AUC del 77 %. No obstante, sigue presentando limitaciones en métricas como la sensibilidad y medida F1.

Para finalizar, el mapa de calor que presenta los mejores resultados de este trabajo es el tercero (4.3c). Aquí se observa que, aunque los algoritmos de Redes Neuronales siguen apareciendo en el top 5, el modelo que mejor rendimiento tuvo fue nuevamente el del algoritmo de Bosques Aleatorios. Este modelo presenta un rendimiento de la métrica ROC-AUC del 86 % y, de forma general, presenta una mejora del 26 % en comparación con la Red Neuronal de Perceptrón Multicapa del primer mapa. Este resultado demuestra que, aunque el algoritmo de Bosques Aleatorios no es tan innovador como una Red Neuronal, sigue mostrando muy buen desempeño para identificar patrones intrínsecos en los datos de forma robusta.

En resumen, los experimentos con datos sintéticos demuestran que el algoritmo con mejor desempeño es el de Bosques Aleatorios, incluso frente a algoritmos más complejos e innovadores como las Redes Neuronales. Este hallazgo refuerza la importancia de no menospreciar algoritmos tradicionales en experimentos de inteligencia artificial, pues técnicas más innovadoras son las que suelen llevarse los reflectores por el mero hecho de ser nuevas.

Los resultados obtenidos permiten formular algunas interpretaciones relevantes sobre el comportamiento de los modelos. En el caso del algoritmo de bosques aleatorios, su ventaja consistente sobre modelos más complejos puede atribuirse a su naturaleza de conjunto: al promediar las predicciones de múltiples árboles de decisión entrenados con subconjuntos aleatorios de datos y características, el modelo reduce el riesgo de sobreajuste y logra capturar patrones intrínsecos en los datos de forma más robusta, cualidad especialmente relevante ante señales biomédicas con variabilidad inherente como los electroretinogramas. Por su parte, la mejora progresiva observada al incrementar el volumen de datos sintéticos —de $x1$ a $x10$ y de $x10$ a $x100$ — sugiere que las limitaciones del conjunto original no eran únicamente de desbalance entre clases, sino también de volumen insuficiente para que los modelos podrían generalizar de forma adecuada.

En términos de aplicación, el mejor resultado obtenido —un ROC-AUC del 86 % mediante bosques aleatorios con datos aumentados por un factor de 100— representa un desempeño alentador considerando que se obtiene a partir de electroretinogramas, un método de adquisición no invasivo, rutinario y de menor costo en comparación con el análisis de imágenes de fondo de retina empleado en la literatura revisada. Si bien este resultado aún presenta margen de mejora para su implementación clínica directa —particularmente en métricas como sensibilidad y medida F1, cuya optimización es crítica para minimizar falsos positivos en un contexto médico—, los hallazgos de este trabajo constituyen una prueba de concepto sólida sobre la viabilidad de detectar principios de retinopatía diabética mediante este enfoque alternativo. Desde una perspectiva de integración, el sistema propuesto requiere como insumo principal una señal de electroretinograma procesada, lo cual implica contar con el equipo de adquisición correspondiente y un flujo de preprocesamiento como el descrito en este trabajo. No obstante,

una vez establecida dicha infraestructura, la clasificación mediante el modelo entrenado representa un proceso automatizado, reproducible y de un costo computacional moderado —particularmente en el caso de bosques aleatorios con datos aumentados por un factor de 100—, lo que favorece su viabilidad en entornos clínicos con recursos limitados. Entre las limitaciones técnicas actuales destacan el tamaño reducido del conjunto de datos reales disponible y la ausencia de variables demográficas, factores que deberán atenderse en etapas posteriores para consolidar este enfoque como una herramienta de apoyo diagnóstico confiable.

Conclusiones

El objetivo principal de esta tesis es desarrollar modelos de aprendizaje automatizado y aprendizaje profundo enfocados en la detección temprana de retinopatía diabética, ofreciéndose como una alternativa a los métodos tradicionales empleados para la identificación de esta enfermedad.

Para cumplir este propósito, se realiza un análisis detallado de los conceptos y herramientas fundamentales dentro del marco teórico, con base en los cuales se diseñan los modelos de aprendizaje automatizado y aprendizaje profundo. Este recorrido incluye la explicación de técnicas clave, algoritmos y metodologías necesarias para abordar la problemática planteada.

En el marco teórico se define el concepto de escalogramas el cuál fue utilizado por el algoritmo de Redes Neuronales Convolucionales (RNC) debido a su amplia capacidad para procesar y clasificar imagenes. Esta definición es clave para entender como los modelos desarrollados en este trabajo fueron sometidos a procesos de optimización de hiperparámetros y extracción de características con técnicas como el Análisis de Componentes Principales y el Análisis Discriminante Lineal.

La validación de los experimentos desarrollados consiste en la realización de modelos diseñados para evaluar el impacto de la optimización de hiperparámetros y técnicas de selección de variables. Cada modelo se evalúa utilizando métricas clave definidas en esta tesis (ROC-AUC, Precisión, Exactitud, Sensibilidad, Especificidad y Medida F1), cuyos resultados se reportan mediante mapas de calor (4.2), lo que permite visualizar tendencias en el desempeño de cada configuración. Cabe destacar que, aunque los experimentos con ADL no logran figurar entre los 10 mejores resultados, las configuraciones optimizadas con ACP alcanzan el mejor desempeño, utilizando un modelo configurado con 1000 épocas, 150 iteraciones y una tasa de aprendizaje de 0.00014. Este modelo tiene un desempeño de ROC-AUC del 65%, un resultado aceptable como prueba de concepto, pero insuficiente para su uso práctico en un contexto médico debido a las limitaciones que se observan en las demás métricas de evaluación.

Adicionalmente, se explora el concepto y funcionamiento de Redes Neuronales Generativas Adversarias (GAN, por sus siglas en Inglés), que se emplearon para generar datos sintéticos, los cuales sirven para balancear y aumentar el conjunto inicial de datos. El proceso de generación se describe detalladamente, desde la creación de datos artificiales con base en los datos reales a través de la parte generadora de la red GAN, pasando por la evaluación iterativa de estos datos mediante la parte discriminante de la GAN, hasta su validación a través de pruebas matemáticas, como el cálculo de la raíz del error cuadrático medio o pruebas de distribución. En esta etapa, se generaron tres conjuntos de datos, denominados x1 (clases balanceadas), x10 (datos aumentados en un factor de 10) y x100 (datos aumentados en un factor de 100).

Como parte de los objetivos que se plantean, se implementan una variedad de modelos de aprendizaje automatizado y aprendizaje profundo que incluyen Árboles de Decisión, Bosques Aleatorios, Máquinas de Soporte Vectorial (SVM), Regresión Logística, Redes Neuronales Convolucionales (CNN), Bayes Ingenuo Gaussiano, Redes Neuronales Recurrentes (RNN) y Perceptrón Multicapa, orientados a la clasificación de pacientes sanos y enfermos con principios de retinopatía diabética. Para garantizar la solidez y desempeño de estos modelos, se llevan a cabo procesos detallados de optimización de hiperparámetros, incluyendo técnicas avanzadas como el Análisis de Componentes Principales (ACP) y el Análisis Discriminante Lineal (ADL), además de ajustes iterativos que permitieron identificar combinaciones óptimas de parámetros. Estos procesos de optimización resultan fundamentales para el desempeño de los modelos, especialmente en aquellos entrenados con conjuntos de datos balanceados y aumentados generados mediante Redes Generativas Adversarias (GAN), técnicas que contribuyen al enriquecimiento de la base de datos inicial y promueven una mayor precisión en los resultados obtenidos.

Para los experimentos con datos sintéticos, la validación también se hace mediante mapas de calor en los que se destacan los cinco mejores resultados para cada caso. De forma general, se observa una mejora importante a medida que se incrementan la cantidad de datos sintéticos para entrenar y probar los modelos, con respecto a los experimentos de los escalogramas. Además, el mejor resultado de todos los experimentos presentados en esta tesis corresponde al modelo que utiliza el algoritmo de Bosques Aleatorios con el conjunto de datos sintéticos aumentados por un factor de 100, donde logra un rendimiento (métrica ROC-AUC) del 86 % y una exactitud del 77 %. Este resultado refleja una buena capacidad del modelo para clasificar a los pacientes enfermos y sanos, tomando en cuenta el método médico utilizado (electrorretinogramas).

El orden cronológico de los experimentos que se presentan en esta tesis sigue el mismo esquema descrito en los resultados: primero se trabaja con los escalogramas y después con los datos sintéticos. Si bien los resultados obtenidos son prometedores, existen aún varias vías de investigación que pueden ser exploradas para mejorar el desempeño y ampliar el alcance de este trabajo.

Algunas áreas de investigación futura que tiene este trabajo es el diseño de arquitecturas más complejas para el algoritmo de Redes Neuronales Convolucionales. Estudios como el de Chen et al. (2020) [37], donde se utilizaron Redes Neuronales Generativas Adversarias para aumentar la disponibilidad de datos en el procesamiento de escalogramas acústicos, y el de Jayalakshmy et al. (2021) [38], quienes aplicaron técnicas similares para la clasificación de señales respiratorias logrando una mejora en la precisión de sus modelos del 81.37 % al 98.75 %, sugieren que una configuración adaptada específicamente a las características de los escalogramas podría mejorar los resultados obtenidos en este trabajo. Además, también se podrían aumentar los datos de forma sintética pero, en lugar de ser aplicados a las frecuencias de los electrorretinogramas, sean aplicados a la generación de escalogramas para observar si la tendencia de mejora en los resultados a medida que se incrementa la información, observada en este trabajo, también es aplicable para los escalogramas. Con base en la evidencia de los estudios mencionados, la aplicación de estos enfoques en el objeto de estudio de esta tesis podría facilitar mejoras en las métricas de evaluación de los modelos, especialmente considerando que la disponibilidad de datos es limitada.

Los avances que se logran en esta tesis abren las puertas para futuras investigaciones en el área de aprendizaje automatizado y aprendizaje profundo, sentando las bases para explorar enfoques que mejoren la salud pública mediante herramientas tecnológicas eficientes y accesibles. Los trabajos revisados en la literatura, basados predominantemente en el análisis de imágenes del fondo de retina, reportan niveles de exactitud que oscilan entre el 78 % [18] y el 98.45 % [14]. El mejor resultado obtenido en esta tesis,

con un ROC-AUC del 86 % mediante el algoritmo de Bosques Aleatorios, se sitúa dentro de ese rango, aunque empleando un método de adquisición de datos fundamentalmente distinto y menos invasivo: los electroretinogramas. Si bien aún existe margen de mejora, estos resultados son alentadores considerando que los métodos propuestos se distinguen por ser más rápidos, más accesibles y potencialmente más cómodos para el paciente, lo que resulta prometedor para su implementación en entornos clínicos con recursos limitados.

Por último, esta tesis representa un aporte para el desarrollo de alternativas en la clasificación de pacientes con retinopatía diabética de una forma más accesible, pues la aplicación de nuevas técnicas y enfoques persigue una misión muy noble: apoyar en el diagnóstico de personas que tienen el riesgo de desarrollar esta enfermedad. La combinación de las técnicas presentadas en este trabajo (el uso de escalogramas y la generación de datos sintéticos) contribuye a la construcción de un enfoque más robusto y versátil para abordar la problemática de esta enfermedad.

Bibliografía

- [1] V. Rajaraman, “John mccarthy — father of artificial intelligence,” *Resonance*, vol. 19, pp. 198–207, 2014.
- [2] E. Alpaydin, *Introduction to machine learning*. MIT Press, 2020.
- [3] R. Noguez Imm, J. Muñoz-Benitez, D. Medina, E. Bárcenas, G. Molero-Castillo, P. Reyes-Ortega, J. A. Hughes-Cano, L. Medrano-Gracia, M. Miranda-Anaya, G. Rojas-Piloni, *et al.*, “Preventable risk factors for type 2 diabetes can be detected using noninvasive spontaneous electroretinogram signals,” *PLoS ONE*, vol. 18, no. 1, p. e0278388, 2023.
- [4] I. Federation, “International diabetes federation diabetes atlas, 10th edn,” 2021.
- [5] L. Moreno-Altamirano, J. García-García, G. Soto-Estrada, S. Capraro, and D. Limón-Cruz, “Epidemiología y determinantes sociales asociados a la obesidad y la diabetes tipo 2 en México,” *Revista Médica del Hospital General de México*, vol. 77, no. 3, pp. 114–123, 2014.
- [6] S. Naik, D. Kamidi, S. Govathoti, R. Cheruku, and A. Mallikarjuna Reddy, “Efficient diabetic retinopathy detection using convolutional neural network and data augmentation,” *Soft Computing*, pp. 1–12, 2023.
- [7] P. Mendes, M. Rocha, *et al.*, “Synthetic patient data generation and evaluation in disease prediction using small and imbalanced datasets,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4747–4756, 2022.
- [8] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
- [9] B. M. Ege, O. K. Hejlesen, O. V. Larsen, K. Møller, B. Jennings, D. Kerr, and D. A. Cavan, “Screening for diabetic retinopathy using computer-based image analysis and statistical classification,” *Computer Methods and Programs in Biomedicine*, vol. 62, no. 3, pp. 165–175, 2000.
- [10] A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, “Automatic recognition of exudative maculopathy using fuzzy c-means clustering and neural networks,” in *Proc. Medical Image Understanding Analysis Conf.*, vol. 3, pp. 49–52, Citeseer, 2001.
- [11] M. Larsen, J. Godt, N. Larsen, H. Lund-Andersen, A. K. Sjølie, E. Agardh, H. Kalm, M. Grunkin, and D. R. Owens, “Automated detection of fundus photographic red lesions in diabetic retinopathy,” *Investigative Ophthalmology & Visual Science*, vol. 44, no. 2, pp. 761–766, 2003.

-
- [12] H. Narasimha-Iyer, A. Can, B. Roysam, V. Stewart, H. L. Tanenbaum, A. Majerovics, and H. Singh, “Robust detection and classification of longitudinal changes in color retinal fundus images for monitoring diabetic retinopathy,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1084–1098, 2006.
- [13] M. Niemeijer, B. van Ginneken, S. R. Russell, M. S. Suttorp-Schulten, and M. D. Abramoff, “Automated detection and differentiation of drusen, exudates, and cotton-wool spots in digital color fundus photographs for diabetic retinopathy diagnosis,” *Investigative Ophthalmology & Visual Science*, vol. 48, no. 5, pp. 2260–2267, 2007.
- [14] A. G. Karegowda, A. Nasiha, M. Jayaram, and A. Manjunath, “Exudates detection in retinal images using back propagation neural network,” *International Journal of Computer Applications*, vol. 25, no. 3, pp. 25–31, 2011.
- [15] J. De la Calleja, L. Tecuapetla, M. Auxilio Medina, E. Bárcenas, and A. B. Urbina Nájera, “Lbp and machine learning for diabetic retinopathy detection,” in *Intelligent Data Engineering and Automated Learning–IDEAL 2014: 15th International Conference, Salamanca, Spain, September 10–12, 2014. Proceedings 15*, pp. 110–117, Springer, 2014.
- [16] S. Roychowdhury, D. D. Koozekanani, and K. K. Parhi, “Dream: diabetic retinopathy analysis using machine learning,” *IEEE journal of biomedical and health informatics*, vol. 18, no. 5, pp. 1717–1728, 2013.
- [17] M. T. Hagos and S. Kant, “Transfer learning based detection of diabetic retinopathy from small dataset,” *arXiv preprint arXiv:1905.07203*, 2019.
- [18] G. T. Reddy, S. Bhattacharya, S. S. Ramakrishnan, C. L. Chowdhary, S. Hakak, R. Kaluri, and M. P. K. Reddy, “An ensemble based machine learning model for diabetic retinopathy classification,” in *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pp. 1–6, IEEE, 2020.
- [19] Z. Khan, F. G. Khan, A. Khan, Z. U. Rehman, S. Shah, S. Qummar, F. Ali, and S. Pack, “Diabetic retinopathy detection using vgg-nin a deep learning architecture,” *IEEE Access*, vol. 9, pp. 61408–61416, 2021.
- [20] M. Scarpiniti, R. Parisi, and Y.-C. Lee, “A scalogram-based cnn approach for audio classification in construction sites,” *Applied Sciences*, vol. 14, no. 1, p. 90, 2023.
- [21] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.
- [22] Oracle, “Imagen sobre áreas relacionadas con IA,” 2023. Disponible en: <https://www.oracle.com/a/ocom/img/cc01-what-is-artificial-intelligence-diagram.png>. Consultado: Octubre 2025.
- [23] G. Rebalá, A. Ravi, and S. Churiwala, *An introduction to machine learning*. Springer, 2019.
- [24] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd, 2019.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
-

-
- [26] D. A. Roberts, S. Yaida, and B. Hanin, *The principles of deep learning theory*, vol. 46. Cambridge University Press Cambridge, MA, USA, 2022.
- [27] X. B. Olabe, “Redes neuronales artificiales y sus aplicaciones,” *Publicaciones de la Escuela de Ingenieros*, vol. 5, 1998.
- [28] R. A. García and A. F. Lepera, “El perceptrón multicapa: Modelo matemático e implementación en python,” *Inf. téc. Facultad de Ciencias Económicas, Universidad de Buenos Aires*, 2020.
- [29] C. Arana, “Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales,” tech. rep., Serie Documentos de Trabajo, 2021.
- [30] M. M. Taye, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” *Computation*, vol. 11, no. 3, p. 52, 2023.
- [31] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024.
- [32] P. Calvo, “Esquema de red neuronal convolucional,” 2017. (Consultado: Noviembre, 2025).
- [33] J. Niu, Y. He, M. Li, X. Zhang, L. Ran, C. Chao, and B. Zhang, “A comparative study on application of data mining technique in human shape clustering: Principal component analysis vs. factor analysis,” in *2010 5th IEEE Conference on Industrial Electronics and Applications*, pp. 2014–2018, IEEE, 2010.
- [34] C. Studio, “Imagen generada con inteligencia artificial: Representación gráfica del análisis de pca,” 2025.
- [35] S. Balakrishnama and A. Ganapathiraju, “Linear discriminant analysis-a brief tutorial,” *Institute for Signal and information Processing*, vol. 18, no. 1998, pp. 1–8, 1998.
- [36] M. Hossin and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
- [37] H. Chen, Z. Liu, Z. Liu, and P. Zhang, “Acgan-based data augmentation integrated with long-term scalogram for acoustic scene classification,” *arXiv preprint arXiv:2005.13146*, 2020.
- [38] S. Jayalakshmy and G. F. Sudha, “Respiratory signal classification by cgan augmented emd-scalograms,” in *2021 IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, Communication (AESPC)*, pp. 1–5, 2021.