Predicción del desenlace terapéutico de leishmaniasis cutánea con base en fotografías de lesiones e información del transcriptoma.

Karen Andrea Acevedo
Mario Arrieta Sánchez
Catalina Gómez Vallejo

Anteproyecto del Proyecto Aplicado para optar al título de Magister en Ciencia de Datos

Director(a)
Diego Linares

Codirector(a)
María Adelaida Gómez

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, ENERO 15 DE 2024

Nota Aceptación Trabajo de Grado

Predicción del desenlace terapéutico de leishmaniasis cutánea con base en fotografías de lesiones e información del transcriptoma.

Karen Andrea Acevedo

Mario Arrieta Sánchez

Catalina Gómez Vallejo

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.

Diego Linares - director

Julian Gil Gonzalez - jurado

Omar Andres Castano - jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en Ciencia de Datos.

HERNÁN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias

JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Santiago de Cali 11 de marzo de 2024

Carta solicitud sustentación Trabajo de Grado

Santiago de Cali, 23 de Abril de 2024

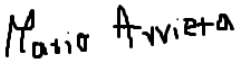
Ingeniero:
Juan Carlos Martínez Arias
Director Posgrados de Ingeniería
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana - Cali

Con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el Trabajo de Grado y posteriormente optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto de Trabajo de Grado denominado predicción del desenlace terapéutico de leishmaniasis con base en fotografías de lesiones e información del transcriptoma, el cual será realizado por los estudiantes Karen Andrea Acevedo, Mario Arrieta Sánchez y Catalina Gómez Vallejo con código 8975294, 8975725 y 8974333 perteneciente al énfasis en Sistemas y Computación, bajo la dirección del profesor Diego Linares.

El suscrito director del Trabajo de Grado autoriza para que se proceda a hacer la evaluación de este Proyecto ante el Tribunal que para el efecto se designe, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado oficialmente.

Atentamente,

Karen Andrea Acevedo
C.C. 1019109777 de Bogotá



Mario Arrieta Sánchez
C.C. 110401519 de San Pedro, Sucre

Catalina Gómez Vallejo
C.C. 1.143.850.459 de Cali

Diego Linares
C.C. Cédula. de Ciudad

Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias

FICHA RESUMEN
TRABAJO DE GRADO DE MAESTRÍA

TITULO: " Predicción del desenlace terapéutico de leishmaniasis con base en fotografías de lesiones e información del transcriptoma."

1. ÉNFASIS: Sistemas y Computación
2. TIPO DE PROYECTO: Aplicado
3. ÁREA DE TRABAJO: Sector Salud - CIDEIM
4. ESTUDIANTE (S): Catalina Gómez, Karen Acevedo, Mario Arrieta.
5. CORREO ELECTRÓNICO: catagova@javerianacali.edu.co, 1012karen@javerianacali.edu.co, mariojarrieta@javerianacali.edu.co
6. DIRECCIÓN Y TELÉFONO: Carrera 40a #13b-48 Cali - 3053787086, Calle 104 #57a-22 Bogotá - Colombia – 3192277999, Calle 12 #14-39 San Pedro, Sucre – 3225740227
7. DIRECTOR: Diego Linares
8. VINCULACIÓN DEL DIRECTOR (en la universidad): Planta
9. CORREO ELECTRÓNICO DEL DIRECTOR: dlinares@javerianacali.edu.co
10. CO-DIRECTOR(ES) (Si aplica): María Adelaida Gómez
11. GRUPO O EMPRESA QUE LO AVALA (Si aplica): CIDEIM
12. OTROS GRUPOS O EMPRESAS:
13. PALABRAS CLAVE: Leishmaniasis, Desenlace Terapéutico, Redes Neuronales, Predicción, Ciencia de Datos, procesamiento de imágenes, ANOVA, Eliminación Recursiva de Características (RFE), Análisis de componentes principales (PCA).
14. ODS 9: Industria, innovadora e infraestructura (Agenda 2030):
15. FECHA DE INICIO (Desarrollo del proyecto): 1/01/2023

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTA DE TABLAS

RESUMEN

Esta investigación adoptó un enfoque cuantitativo de carácter descriptivo-experimental, en el cual se utilizó una metodología centrada en la recopilación y análisis de datos numéricos e imágenes para describir detalladamente las variables de interés. Este método se distingue por su énfasis en la medición objetiva de las variables mediante un diseño experimental a partir del conjunto de datos disponible.

El proyecto se desarrolló utilizando fotografías de lesiones y datos de información transcriptómica de un grupo de pacientes que previamente habían sido tratados por el CIDEIM, con el propósito de evaluar la eficacia del tratamiento para la leishmaniasis. Este enfoque incorporó herramientas de aprendizaje automático donde se requirió la construcción de bases de datos de alta calidad para llevar a cabo el procesamiento, la aplicación de las técnicas y su evaluación.

Después de la creación de los conjuntos de datos e imágenes, se aplicaron técnicas esenciales en la preparación de datos tanto para los transcriptomas como para las imágenes, con el objetivo de mejorar la calidad y simplificar el análisis. En el caso de los datos de transcriptomas, se comenzó aplicando técnicas de limpieza y reducción de dimensionalidad, como ANOVA, PCA y RFE, que permitieron segmentar y extraer los genes más significativos para cumplir con los objetivos establecidos.

Posteriormente, se implementaron modelos de aprendizaje supervisado, tales como SVM, Árboles de Decisión, K-vecinos y Bosques Aleatorios. Estos modelos fueron evaluados mediante un conjunto de entrenamiento aplicando validación cruzada, con el propósito de analizar tanto los modelos base como aquellos que resultaron de la estimación de los mejores hiperparámetros, buscando alcanzar un rendimiento óptimo. La evaluación del desempeño de estos modelos se llevó a cabo a través del conjunto de prueba, verificando los resultados frente a pruebas de laboratorio de referencia.

Se analizaron diversas métricas, como sensibilidad y especificidad, con el objetivo de evaluar la coherencia entre los métodos, y se evidenció un rendimiento generalmente satisfactorio. No obstante, al emplear los genes seleccionados mediante el método de ANOVA, se destacó una consistencia notable tanto en los modelos base como en los estimados. En este escenario, se logró un promedio de exactitud del 0.80 y un F1 score de aproximadamente 0.73 para los modelos base. Tras la estimación de los mejores hiperparámetros, se observó un incremento de alrededor del 0.05 en exactitud y un aumento de 0.07 en el F1 score.

El conjunto de imágenes, por su parte, fue sometido a técnicas como las redes neuronales, para analizar las características particulares, como texturas, formas, bordes y coloración. Esto posibilitó la detección y clasificación automática de los individuos entre cura o falla (no cura). Para abordar esto, se creó un modelo utilizando un conjunto de entrenamiento aplicando validación cruzada, donde se planteó una red neuronal base a la cual se le realizó una estimación de hiperparámetros para obtener el mejor rendimiento. Posterior se utilizaron las arquitecturas VGG16 y VGG19 junto con la transferencia de aprendizaje de los hiperparámetros definidos de la red base. La evaluación del desempeño de estos modelos se llevó a cabo a través de conjuntos de prueba obteniendo con estas dos arquitecturas (VGG16 y VGG19) los resultados óptimos. Una exactitud promedio de 0.92 y una función de pedida promedio de 0.17.

En última instancia, se realizó la integración de los mejores modelos analizados. Comprendió cinco modelos supervisados, seleccionados en función de su rendimiento según las métricas previamente mencionadas. Con esta integración, se llevaron a cabo pruebas utilizando los transcriptomas y las fotografías de lesiones de dos pacientes que hacían parte del conjunto de prueba para determinar si se clasificaban adecuadamente como cura o falla (no cura).

Se observó que, para el primer paciente, todos los modelos coincidieron en clasificarlo como falla (no cura), lo cual resultó ser correcto. En el caso del segundo paciente, tres de los cinco modelos lo catalogaron como cura, mientras que los dos restantes lo clasificaron como falla (no cura). A pesar de esta discrepancia, debido a que tres de los cinco modelos acertaron al clasificarlo como cura, el paciente fue finalmente clasificado dentro de la categoría adecuada.

A partir de los resultados obtenidos, fue posible reconocer y extraer características significativas tanto de los genes como de las imágenes, las cuales sirvieron como indicadores morfológicos de la presencia de leishmaniasis cutánea en el individuo. En última instancia, se realizó la interpretación de los resultados obtenidos para evaluar la viabilidad del proyecto, identificando limitaciones y desafíos, así como posibles cambios y mejoras para futuras investigaciones.

INTRODUCCIÓN

La leishmaniasis cutánea (LC) constituye una enfermedad endémica presente en más de 88 países, siendo dos tercios de los casos registrados en naciones como Afganistán, Argelia, Brasil, Pakistán, Perú, Arabia Saudita, Irán y Siria [1,2]. Esta patología muestra una prevalencia anual de 0,7-1,2 millones de casos a nivel mundial [3]. En Colombia, la LC representa un significativo problema de salud pública, evidenciado por el incremento en el número de casos reportados y la emergencia de nuevas cepas que presentan resistencia a los principales medicamentos empleados en el tratamiento, especialmente los antimoniales. Además, se suma la creciente preocupación respecto a la falla terapéutica, que ha llevado a proporciones epidémicas en diversas regiones del mundo. A inicios de los 2000, en la India se estimaba que entre el 35% al 65% de los pacientes con leishmaniasis no respondían al tratamiento [4]. América Latina, en países como Brasil también se experimentaba una tasa de falla en respuesta al tratamiento del 25-50%, para Perú alrededor del 24% y en Colombia estaba entre el 15-39% respectivamente [5]. Actualmente estos tres países reportan el 85% de los casos de LC [60].

Ante este desafío, es ampliamente reconocido que la respuesta al tratamiento es un fenómeno multifactorial. Por ende, la identificación de los mecanismos que conducen a la falla en el tratamiento de la LC se ha convertido en un reto crucial. La búsqueda de soluciones ha llevado a la adopción de diversas tecnologías, destacándose el aprendizaje automático por su capacidad para analizar grandes volúmenes de información y descubrir patrones relevantes.

Partiendo de lo expuesto y conscientes de la complejidad del problema, el enfoque de la presente investigación radica en analizar el desenlace terapéutico de la leishmaniasis cutánea (LC), empleando información proveniente de lesiones y transcriptomas de pacientes curados y no curados (falla). Se utilizarán técnicas avanzadas de ciencia de datos con el objetivo de construir una serie de modelos que faciliten la predicción de la eficacia del tratamiento en pacientes con ciertas características específicas.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La leishmaniasis forma parte de las enfermedades parasitarias de transmisión vectorial. Según el Ministerio de Salud, se conocen al menos 20 especies de parásitos del género Leishmania. Esta enfermedad se transmite a animales y seres humanos a través de la picadura de insectos de la familia Psychodidae, principalmente insectos flebótomos hembras. La infección por Leishmania en humanos puede manifestarse de tres formas: leishmaniasis cutánea, mucosa y visceral [6]. Su prevalencia es más notable en áreas rurales que afectan a poblaciones vulnerables [7].

La enfermedad según sea el caso puede presentar síntomas como úlceras no dolorosas de bordes elevados, fiebre, aumento de tamaño del abdomen, pérdida de apetito, disminución de peso, tos seca, diarrea y vómitos [8]. El diagnóstico es clínico, complementado por pruebas parasitológicas, histológicas o inmunológicas específicas. Uno de los métodos de investigación usados para analizar las características de las lesiones es el transcriptoma, el cual es una colección de todas las lecturas de genes presentes en una célula, utilizando muestras de lesiones en la piel (placa de biopsia). Dada la posibilidad de que estos síntomas se asocien con otras afecciones cutáneas, el diagnóstico puede ser desafiante, y las técnicas convencionales pueden presentar limitaciones en términos de sensibilidad, especificidad y facilidad de uso.

El diagnóstico oportuno es fundamental para iniciar un tratamiento adecuado y mejorar la calidad de vida de los pacientes. Sin intervención, las formas mucosa y cutánea pueden causar deformidades, y la variante visceral puede ser letal en más del 90% de los casos no tratados [6]. El tratamiento principal implica el uso de fármacos. No obstante, es importante tener en cuenta que estos tratamientos pueden ocasionar efectos adversos y su efectividad no está garantizada.

Dada la complejidad de identificar patrones en las fotografías de las lesiones y transcriptomas para obtener un panorama claro del estado de los pacientes, se busca utilizar modelos de aprendizaje automático. Con información de un grupo de pacientes curados y no curados otorgada por el CIDEIM, donde se pretende predecir la efectividad del tratamiento, incorporando así avances de la medicina y la ciencia de datos para mejorar las decisiones clínicas.

1.2. FORMULACIÓN DEL PROBLEMA

Las opciones de prevención y control contra la leishmaniasis son limitadas, lo que hace necesario que las autoridades de salud lleven a cabo múltiples acciones para tratar a los individuos afectados. Las opciones de tratamiento conllevan a la administración de medicamentos tóxicos y con una baja tolerancia en los pacientes [61].

Uno de los principales tratamientos es con sales de antimoniato pentavalente. Otra alternativa para uso en niños mayores a 2 años es el miltefosine, el cual es el primer medicamento de uso oral, este presenta mejor tolerancia en los pacientes, aunque con toxicidad hepática, renal y reacciones adversas gastrointestinales. Estos tratamientos han presentado una eficacia de alrededor del 25% para antimonio y 69% para miltefosine [62]. Adicional, hay otros tratamientos como Anfotericina B o la Pentamidina sin embargo son más costos e igualmente presentan toxidad [63].

Acorde a algunos estudios, la eficacia del tratamiento para la LC varía según múltiples factores como el grupo de edad, la especie parasitaria, la localización de las lesiones y el número, el estado nutricional entre otros [61]. Por lo cual, se constituye como gran reto el determinar si la respuesta al tratamiento para un paciente resultara en falla o en cura.

1.3. PREGUNTAS DE INVESTIGACIÓN

¿Cómo predecir a partir de fotografías de las lesiones y la información del transcriptoma, que tan eficaz sería el tratamiento para la leishmaniasis?

Preguntas de sistematización:

- ¿Cómo procesar la información para que los modelos sean adecuados?
- ¿Cómo modelar el problema usando aprendizaje automático?
- ¿Cómo optimizar el desempeño de los modelos de aprendizaje automático?
- ¿Cómo evaluar la calidad de los modelos desarrollados?
- ¿Cómo utilizar el modelo desarrollado como apoyo al tratamiento?

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Predecir el desenlace terapéutico de la leishmaniasis cutánea con base en fotografías de lesiones e información del transcriptoma, usando modelos de aprendizaje automático.

2.2. OBJETIVOS ESPECÍFICOS

- Preparar las fotografías de lesiones y la información del transcriptoma para que sea útil para los modelos.
- Modelar los datos relacionados al desenlace terapéutico.
- Optimizar los modelos de aprendizaje automático.
- Evaluar el ajuste y la capacidad predictiva de los modelos.
- Evaluar a partir de los modelos los posibles resultados del tratamiento en un paciente.

3. MARCO TEÓRICO Y ANTECEDENTES

3.1 DEFINICIONES

Leishmaniasis:

La leishmaniasis engloba un conjunto de enfermedades provocadas por diversas especies del protozoo flagelado leishmania. Este microorganismo es un parásito intracelular obligado tanto para humanos como para otros mamíferos que ocasiona lesiones a nivel visceral, mucoso y cutáneo [9, 5]. Esta enfermedad, considerada una zoonosis, afecta tanto a seres humanos como a otras especies de mamíferos siendo los caninos los principales reservorios domésticos [10]. Además, animales silvestres como liebres, zarigüeyas, coatíes y jurumíes también actúan como reservorios del parásito [11].

La leishmaniasis se transmite en países tropicales, la mayoría en vía de desarrollo. Pero también se está expandiendo al mediterráneo europeo y a USA. El acceso a tratamientos adecuados suele ser complicado y puede acarrear efectos secundarios adversos. En el continente americano, la leishmaniasis se manifiesta principalmente como una zoonosis selvática (aunque puede adquirirse en regiones semidesérticas o frías) transmitida por flebótomos, principalmente de los géneros Phlebotomus y Lutzomyia [12].

Las características clínicas varían en función de las propiedades del parásito y los aspectos genéticos del huésped los cuales determinan la efectividad de la respuesta inmune. Según las manifestaciones clínicas, la leishmaniasis puede dividirse en tres formas: cutánea, mucosa y visceral [13].

- Leishmaniasis visceral:

La leishmaniasis visceral o kala-azar (término que significa enfermedad negra y que es debido a la coloración grisácea de la piel que es adquirida por habitantes de la India con esta enfermedad) es producida por especies del complejo de leishmania donovani. El protozoo se localiza preferentemente en el sistema mononuclear fagocítico y la enfermedad se caracteriza por un cuadro febril crónico con esplenomegalia progresiva e hipergammaglobulinemia [9, 11].

- Leishmaniasis Mucosa:

La leishmaniasis mucosa ocurre principalmente en las Américas y afecta principalmente la mucosa nasal. Tiene un curso progresivo y puede causar destrucción, deformidad y mutilación en casos severos. La leishmaniasis mucosa es causada por especies de Leishmania del subgénero

Viannia. Esta tiene un curso progresivo y puede causar deformidad e incluso mutilación de las zonas afectadas [14].

- Leishmaniasis cutánea:

Se caracteriza por lesiones cuyo espectro va desde pápulas, placas, ulceras y nódulos localizados en cualquier área de la superficie corporal. Esta manifestación clínica se puede presentar como una leishmaniasis cutánea Localizada (LCL), leishmaniasis cutánea difusa (LC-Difusa), leishmaniasis cutánea diseminada (LC-Diseminada) o leishmaniasis cutánea atípica (LC-Atípica) [15].

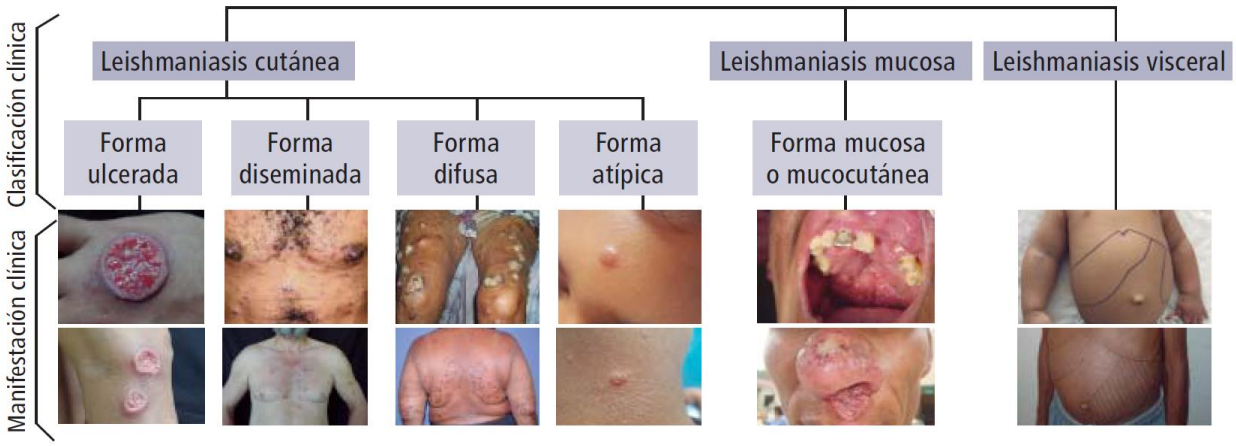


Figura 1

Ilustración 1: ***Tipos de leishmaniasis.***

Fuente: Organización Panamericana de la Salud. Manual de Procedimientos para vigilancia y control de leishmaniasis en las Américas.2019

ADN:

El ADN contiene la información genética única de cada organismo. La Sociedad Española de Médicos de Atención Primaria (SEMERGEN) destaca en un artículo sobre la estructura y función del ADN y los genes que este coordina las interacciones en el funcionamiento celular y tisular. El código genético guía las actividades celulares, logrando un equilibrio entre las influencias ambientales y la red funcional del ADN para producir respuestas a cambios en el entorno. Sin embargo, la capacidad homeostática del genoma puede alterarse por agentes ambientales, resultando en efectos adversos y patológicos [16].

Código Genético:

El ADN transporta la información genética a través de combinaciones de bases. Cada secuencia de tres bases, conocida como codón, determina la codificación de un aminoácido. Por ejemplo, el codón ATG especifica la metionina y marca el inicio de la lectura para el ARN mensajero (ARNm). Este ARNm luego copia el mensaje genético y lo transporta al citoplasma, donde se sintetiza la proteína correspondiente a cada gen. A pesar de que inicialmente se estimaba que existían alrededor de 100,000 proteínas, los avances tecnológicos han ajustado esta cifra a unos 25,000 genes [16].

Transcriptoma:

El genoma humano, compuesto de ADN, contiene las instrucciones para producir y mantener células. Estas instrucciones, expresadas como "pares de bases", requieren ser leídas y transcritas en ARN (ácido ribonucleico) para llevarse a cabo. Las transcripciones génicas se denominan transcritos, y un transcriptoma es una colección de todas las lecturas génicas presentes en una célula [17].

La secuenciación de alto rendimiento (RNA-seq) analiza el transcriptoma y detecta cambios significativos en la expresión génica. Es preferible a otros enfoques de perfiles transcripcionales, como los microarreglos de ADN, y se utiliza en estudios para comprender la interacción entre el parásito y los huéspedes, así como para analizar la progresión del ciclo de vida y comparar entornos naturales en cada etapa [15].

Diagnóstico:

El diagnóstico de la leishmaniasis se establece mediante la observación directa de amastigotes en muestras clínicas, ya sea al microscopio o mediante técnicas moleculares que amplifican el ADN nuclear o del cinetoplasto. Los amastigotes tienen una forma característica en bastón, llamada cinetoplasto, y miden aproximadamente de 1 a 4 µm de diámetro. La sensibilidad de las técnicas de diagnóstico puede ser limitada, por lo que a veces se requiere la combinación de varias técnicas y la toma de múltiples muestras para un diagnóstico preciso [8].

Tratamiento:

Una vez establecido el diagnóstico, se busca revertir el estado patológico detectado en el paciente y limitar el daño. La intervención médica se decide según la historia clínica, estudios paraclínicos y la evaluación de la evolución natural de la enfermedad. En el caso de la leishmaniasis, existen alternativas terapéuticas, ya que no hay una vacuna eficaz. Aunque se

conocen alrededor de 25 compuestos con efecto anti-leishmanial, solo algunos se utilizan en humanos [4].

Ciencia de Datos:

La Ciencia de Datos es un campo multidisciplinario que busca extraer conocimiento de grandes volúmenes de datos, ya sean estructurados o no estructurados. Incluye tecnologías como bases de datos, Big Data que se relaciona no solo con la cantidad masiva de datos, sino también con la variedad, velocidad y acceso a la información, aprendizaje automático e internet de las cosas. Estas herramientas transforman datos en información, permitiendo razonamiento y comprensión para optimizar la toma de decisiones. Las Tecnologías de Información y Comunicaciones (TIC) facilitan el procesamiento y la transformación de datos en decisiones de valor [19].

Aprendizaje Automático:

El aprendizaje automático es una rama de la inteligencia artificial que posibilita que las computadoras aprendan sin programación explícita. Se define como un programa que mejora su rendimiento en una tarea específica a medida que acumula experiencia. La ciencia de programar computadoras para aprender a partir de datos ha evolucionado rápidamente con el crecimiento de datos disponibles, dividiéndose en categorías como aprendizaje supervisado, no supervisado y por refuerzo [20].

3.2 ANTECEDENTES

Para el desarrollo del proyecto se revisaron multiples investigaciones donde se implementaron modelos y tecnicas de aprendizaje automatico en el ambito de la medicina, en su mayoria enfocados a la leishmaniasis que permitieron amplicar e identificar multiples posibilidades. A continuación hacemos referencia algunas de ellas:

El propósito de la investigación [21], que aplica técnicas de selección de características en bioinformática, es proponer alternativas para la identificación de medicamentos utilizables en el tratamiento de la leishmaniasis. En este estudio, se utilizan dos modelos derivados del Aprendizaje Profundo: el DeepPurpose y el MONN.

Se emplea DeepPurpose para la creación de una biblioteca de aprendizaje profundo diseñada para la predicción de interacciones fármaco-objetivo ("A deep learning library for drug–target interaction prediction"). Esta biblioteca utiliza herramientas de aprendizaje profundo para el modelado y la predicción molecular, facilitando la anticipación de interacciones entre candidatos

a fármacos y proteínas relacionadas con la leishmaniasis. Este enfoque permite seleccionar los candidatos más prometedores mediante técnicas de aprendizaje profundo y métodos tradicionales, para su análisis subsiguiente. Por otro lado, MONN, un modelo de aprendizaje profundo se empleó para predecir la afinidad de unión entre proteínas y ligandos. Este modelo utiliza estructuras de unión 3D conocidas para prever interacciones no covalentes entre un par específico, mediante un codificador CNN para la proteína y un codificador MPNN para la molécula. Cabe destacar que MONN, en comparación con el enfoque de aprendizaje profundo, impone limitaciones en la cantidad de datos que puede emplear.

Integrando estas herramientas, los autores identificaron varios fármacos potenciales para el tratamiento de la leishmaniasis mediante la identificación de compuestos susceptibles a ser reutilizados. Se llevaron a cabo comparaciones entre diversos modelos y evaluaciones finales de puntuación, además de un análisis de agrupamiento basado en las propiedades moleculares [21].

En el artículo de Guyón et al. [22], se exploró el uso del método SVM RFE (eliminación recursiva de características) en la selección de genes para el diagnóstico del cáncer de colon. SVM RFE demostró su capacidad para eliminar genes mejor clasificados sin problemas, incluyendo aquellos relacionados con la composición del tejido muscular. Este método, basado en el mecanismo del vector de soporte, se destacó como el único probado que elimina eficazmente y verifica la relevancia de los siete genes mejor clasificados en el diagnóstico del cáncer de colon.

Bamorovat et al. [23], implementaron un enfoque innovador para el diagnóstico y pronóstico de pacientes con leishmaniasis cutánea antroponótica que no respondían al tratamiento. Implementaron redes neuronales artificiales, en particular la estructura de perceptrón multicapa (MLP) de la clase ANN, en conjunto con dos versiones del método de cuantificación del vector de aprendizaje (LVQ), perteneciente a la categoría de ANN. Además, se utilizaron la Máquina de Soporte Vectorial (SVM) y los k vecinos más cercanos (KNN) para realizar una comparación efectiva en esta tarea de clasificación. Los resultados revelaron que el clasificador que logró la mayor precisión fue el SVM, con un 88%, seguido por el MLP, con un 87.8%.

Los datos se recopilaron de historias clínicas y registros demográficos de pacientes, considerando variables como la condición de vivienda, edad, sexo, nivel educativo, duración de la lesión, cantidad y ubicación de las lesiones, progreso del tratamiento y antecedentes de enfermedades crónicas subyacentes. De todas estas características, se determinó que la duración de la lesión tuvo la mayor influencia, ya que su exclusión condujo al índice de precisión más bajo (66.9%). En contraste, el sexo se reveló como el atributo con menor eficacia en la

prueba [23].

El estudio presentado por Shabanpour et al. [24] se centra en la integración de algoritmos de aprendizaje automático con el objetivo principal de prever espacialmente la incidencia de la leishmaniasis cutánea en la provincia de Isfahan, Irán. Se utilizaron tres algoritmos de aprendizaje automático: árbol de decisión (DT), regresión de vector de soporte (SVR) y regresión lineal (LR).

Se recopilaron datos espaciales sobre la incidencia de la enfermedad en la provincia de Isfahan entre los años 2011 y 2018, junto con datos ambientales que incluyeron temperatura, humedad, lluvia, altitud, pendiente, velocidad del viento, índice de vegetación de diferencia normalizada (NDVI), número de días soleados, número de días con heladas y distancia al arroyo. La metodología comprendió un análisis de multicolinealidad de las variables, un análisis geoestadístico utilizando los índices I de Moran y Gi* de Getis-Ord para explorar la autocorrelación espacial, un análisis de puntos críticos de la leishmaniasis y la aplicación de algoritmos de aprendizaje automático.

Entre los resultados más destacados, se observa que la precisión de los mapas generados con los algoritmos DT, SVR y LR fue de 0,951, 0,934 y 0,914, respectivamente, según la curva característica operativa del receptor (ROC) y el área bajo la curva (AUC). Además, se llegó a la conclusión de que las áreas este y sur de la provincia presentan el menor riesgo de leishmaniasis [24].

De acuerdo con las investigaciones llevadas a cabo por Conde et al. [25], quienes se propusieron desarrollar un sistema de detección de leishmaniasis, centrándose en la creación e implementación de un sistema para la detección temprana de la enfermedad mediante la aplicación de técnicas de inteligencia artificial y el análisis del historial de la población afectada en el departamento del Norte de Santander. El estudio incorporó imágenes de lesiones cutáneas confirmadas por métodos convencionales en pacientes con leishmaniasis cutánea, utilizando herramientas de visión artificial y métodos de inteligencia artificial. A través de técnicas y algoritmos, se extrajo información relevante a partir de las imágenes segmentadas. Como resultado, se propuso el desarrollo de una interfaz web capaz de identificar la enfermedad mediante la toma de imágenes digitales, ofreciendo una posible impresión diagnóstica para la población afectada por esta enfermedad.

Los diversos artículos mencionados facilitan la identificación de diversas técnicas de la ciencia de datos aplicadas al análisis de enfermedades, tanto en el diagnóstico de los pacientes como en su tratamiento. Además, contribuyen a la identificación de características relevantes de las

enfermedades y herramientas de clasificación que son enriquecedoras para abarcar esta investigación.

3.3 MARCO TEÓRICO

Reducción de dimensionalidad:

Análisis de Varianza (ANOVA):

El análisis de varianza (ANOVA) de una vía, utilizado en este proyecto, se basa en la técnica desarrollada por Ronald Aylmer Fisher. En este enfoque, se considera una variable independiente o factor, que es la clase o estatus de los pacientes (cura o falla), y variables dependientes o respuestas, como la expresión génica de los genes. La implementación requiere cumplir con requisitos como homocedasticidad, normalidad, nivel de media de intervalo o superior, y grupos independientes.

El cálculo implica determinar la variabilidad total de los datos y sus componentes, incluyendo la varianza de las medias de los grupos y la varianza dentro de cada grupo (ERROR). El modelo ANOVA supone que, para cada nivel del factor, la respuesta sigue el modelo $Y = \mu i + ERROR$, donde ERROR es una variable aleatoria normal con media 0 y varianza desconocida [26].

El objetivo es verificar si el factor X influye en el valor medio de Y. La hipótesis nula (H_0) y alternativa (H_1) se plantean, evaluando si hay diferencias significativas en la expresión génica entre las clases (cura y falla) para algún gen. Se busca identificar los genes con expresión significativamente diferente entre las clases para un análisis de clasificación posterior.

Si se acepta la hipótesis nula, se espera que la variabilidad dentro de cada grupo sea similar a la variabilidad global. Contrariamente, si se rechaza la hipótesis nula, la variabilidad global es proporcionalmente mayor que la estimada dentro de los grupos.

Se calculan la suma de cuadrados intra e intergrupal, los grados de libertad y las medias de los cuadrados. Se utiliza la estadística F, que se contrasta con el valor crítico F0.95: α = 0.05. Si el resultado indica el rechazo de la hipótesis nula, se concluye que existen diferencias entre los grupos, respaldando la toma de decisiones.

Eliminación Recursiva de Características (RFE) utilizando SVM:

La Eliminación Recursiva de Características (RFE) utilizando SVM asegura una separación efectiva de datos de alta dimensión. La función de decisión para hiperplanos se determina mediante el núcleo lineal, y los valores del vector de ponderación de características se utilizan para evaluar la importancia de estas características. En el caso de varias clases, se calcula el número total de hiperplanos (q) según la ecuación $q = c\,(c - 1)\,/2$ [21]. En el contexto de la función de decisión lineal, el vector x representa las componentes de un espectro, y w es un vector perpendicular al hiperplano.

$$D(x) = sign(x * wj), j = 1,2,3, \dots, q \text{ (1)}$$

En SVM, el límite de decisión se determina a partir de vectores de soporte, y la importancia relativa de las variables se refleja en el vector ponderado. Dado que este vector se sitúa en la ubicación donde el límite de decisión maximiza el margen, un valor elevado en el vector ponderado para una característica específica indica que dicha característica puede separar las clases de manera más distintiva. La ecuación (2) se emplea para calcular el peso, contribuyendo así a la evaluación de la importancia de la variable según SVM-RFE [22].

$$W_s = \frac{1}{q}\sum_{i=1}^{q} Wi \text{ (2)}$$

Análisis de componentes principales (PCA):

El Análisis de Componentes Principales (PCA) es esencial en la reducción de dimensionalidad, permitiendo representar eficientemente la información de un conjunto de datos en un espacio de menor dimensión mientras conserva la mayor cantidad posible de información [27] lo cual lo hace una técnica llamativa para aplicar en la investigación. PCA utiliza la descomposición de valores singulares o la diagonalización de la matriz de covarianza para identificar los componentes principales, los cual implica encontrar los ejes principales que maximizan la varianza de los datos [28, 29]. La elección del número de componentes implica considerar la varianza explicada y la proporción acumulativa [30]. Los pasos para su implementación incluyen la normalización de datos, el cálculo de la matriz de covarianza, la obtención de autovectores y autovalores, la ordenación y selección de componentes, y finalmente la transformación de datos para obtener las coordenadas en el nuevo espacio de características [31].

Modelos:

Máquinas de Soporte Vectorial (SVM):

Las Maquinas de Soporte Vectorial (SVM) son un conjunto de algoritmos de aprendizaje supervisado ampliamente utilizado para clasificación y regresión y por lo tanto es una buena alternativa para el uso en la presente investigación. Utiliza una frontera de decisión alrededor del dominio de los datos y mediante un kernel como el Gaussiano, transforma el espacio de características a una dimensión superior buscando lograr la máxima separación entre las clases. La aplicación del modelo permite entrenar y predecir la clase de nuevas muestras [32].

El SVM separa las clases mediante un hiperplano definido por los puntos más cercanos de dos clases (vectores de soporte), y la función de pérdida se minimiza utilizando el algoritmo de gradiente descendente [32]. La expresión algebraica de SVM se presenta como:

$$f(x) = sign(w'x + b) \text{ (1)}$$

Donde:

- $f(x)$ Es la función de predicción de SVM
- w Es el vector de pesos de la máquina
- x Es el vector de entrada
- b Es el sesgo de la máquina

Árbol de Decisión (DT):

El árbol de decisión es un modelo predictivo que utiliza una estructura jerárquica de nodos para tomar decisiones basadas en características de los datos [34]. Busca aprender una función $f(X) \rightarrow Y$, donde x_i es el conjunto de características y y_i es la variable de respuesta. El algoritmo de los árboles de decisión consta de una función de decisión $D(x)$, un criterio de decisión en cada nodo que determina la rama a seguir, y un modelo construido recursivamente hasta llegar a una hoja. La función de decisión se expresa como la suma de términos ponderados por condiciones booleanas. El entrenamiento del árbol implica encontrar criterios de decisión óptimos en cada nodo, minimizando la impureza o maximizando la ganancia de información, utilizando medidas como la entropía en el caso de clasificación [34].

$$D(x) = \sum_{j=1}^{j} I\left(Q_j(x)\right).C_j \text{ (1)}$$

Donde:

- J es el número total de nodos en el árbol.
- $I(.)$ es una función indicadora que devuelve 1 si la condición es verdadera y 0 de lo contrario.
- C_j es el valor asociado a la hoja j.

Algoritmo de K – vecinos más cercanos (KNN):

K-Nearest Neighbors (KNN) es un método de aprendizaje supervisado utilizado para clasificación y regresión. Emplea un conjunto de datos etiquetado para entrenar y predecir en datos no etiquetados. La hipótesis fundamental es que objetos similares comparten características comunes y tienden a estar en la misma vecindad en el espacio de características [35].

La fórmula matemática de KNN se centra en el cálculo de la distancia euclidiana entre puntos de datos, siendo crucial la elección meticulosa de métricas. En clasificación, asigna etiquetas basándose en la mayoría de votos de sus k vecinos más cercanos, mientras que en regresión estima un valor promedio ponderado [37]. La elección adecuada de k es crítica, afectando la capacidad del modelo para generalizar.

$$d(X_q, X_i) = \sqrt{\sum_{j=1}^{n} (X_{qj} - X_{ij})^2} \quad (1)$$

Donde:

- X_q, es el vector de características de la instancia de prueba.
- X_i, es el vector de características de la i-ésima instancia de entrenamiento.
- n, es el número de características.

KNN destaca por su simplicidad y facilidad de interpretación, pero su rendimiento depende de la calidad del conjunto de datos y las elecciones de diseño [36, 37, 38].

Bosque Aleatorio (Random Forest):

El modelo de Bosque Aleatorio, una técnica de aprendizaje en conjunto basada en árboles de decisión ha demostrado su eficacia en diversas aplicaciones. Este enfoque utiliza la construcción de múltiples árboles, donde cada árbol T_i se entrena con un subconjunto aleatorio de datos y características. La predicción final del Bosque Aleatorio para una nueva observación x se calcula promediando las predicciones individuales de los árboles, como se expresa en la fórmula:

$$Y_{RF}(x) = \frac{1}{N} \sum_{i=1}^{N} Y_{T_i}(x) \quad (1)$$

Donde:

- $Y_{RF}(x)$ es la predicción del bosque aleatorio para x .
- $Y_{T_i}(x)$ es la predicción del árbol $T_i$ipara x .

El Bosque Aleatorio aborda el sobreajuste mediante la diversidad entre los árboles, siendo especialmente útil en conjuntos de datos extensos con muchas características. Además, destaca por su capacidad para manejar datos faltantes sin necesidad de imputación. La elección de hiperparámetros, como el número de árboles, la profundidad máxima y otros, impacta en el rendimiento del modelo [39].

Redes Neuronales Convolucionales (CNN):

Las CNN, han demostrado un éxito significativo en aplicaciones prácticas, especialmente en tareas de visión por computadora, como el reconocimiento de objetos en imágenes. Este tipo específico de red neuronal está diseñado para procesar datos que presentan una estructura en forma de cuadrícula, siendo eficaz para datos como imágenes, que se representan como una cuadrícula 2D de píxeles [40].

La arquitectura de una CNN incluye capas convolucionales, capas de agrupación y capas completamente conectadas, trabajando en conjunto para extraer características jerárquicas y realizar tareas avanzadas como la clasificación de imágenes. Inspiradas en el córtex visual humano, las CNN contienen múltiples capas ocultas especializadas dispuestas en una jerarquía, desde capas iniciales que detectan elementos simples hasta capas más profundas especializadas en reconocer estructuras complejas [40].

Arquitecturas VGG16 y VGG19:

El diseño de las redes neuronales convolucionales VGG16 y VGG19, propuestas por Simonyan y Zisserman de la Universidad de Oxford, destaca en el aprendizaje profundo para reconocimiento de imágenes. Su arquitectura simple incluye capas de convolución, max-pooling, activaciones ReLU y softmax, esenciales para procesar características visuales en imágenes. Ambas utilizan bloques progresivos de capas convolucionales con filtros 3x3 y max-pooling para reducir dimensiones. La fase de clasificación tiene dos capas densas de 4096 neuronas y una capa de salida con 1000 neuronas. La designación "16" o "19" indica el número total de capas entrenables en cada red. Esta información se resume en una tabla comparativa que destaca las diferencias entre VGG19 y VGG16 en el contexto del aprendizaje profundo [41].

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Ilustración 2: ***Arquitectura vgg19 vs vgg16 en Deep Learning***

Fuente: Redacción KeepCoding [55].

Método de ensamble por votación:

Este método de ensamble combina las predicciones de múltiples modelos mediante votación para clasificar nuevos ejemplos, siendo parte de las técnicas conocidas como métodos multiclasificadores o de ensamble [42]. Entre estas técnicas se incluyen votación por mayoría, promedio, ponderación, apilamiento, entre otras más robustas. En este proyecto, se selecciona un método básico de ensamble, la votación, que combina las predicciones de varios clasificadores y predice la clase que recibe la mayoría de los votos.

El enfoque de votación implica la combinación de predicciones de varios modelos individuales para tomar una decisión final, buscando un rendimiento más robusto y generalizado [43]. Existen dos tipos principales de votación, votación dura (hard voting) y votación suave (soft voting) las cuales describiremos en los siguientes ejemplos:

- Votación Dura:

Imaginemos un comité médico que debe tomar decisiones sobre el tratamiento de un paciente con una enfermedad específica. Cada miembro del comité representa un modelo de clasificación diferente basado en sus experiencias y conocimientos. En un enfoque de hard voting, cada miembro emite un voto definitivo sobre el tratamiento recomendado. El tratamiento elegido será aquel que reciba la mayoría de los votos. En este caso, el hard voting es adecuado cuando se busca una decisión clara y única, y se confía en que la mayoría tiene la respuesta correcta.

- Votación Suave:

Supongamos que estamos trabajando en un proyecto de análisis de sentimientos para clasificar reseñas de productos en positivas o negativas. Tenemos tres modelos de clasificación: un modelo de regresión logística, un clasificador de máquinas de soporte vectorial (SVM) y un modelo de bosque aleatorio. En un enfoque de soft voting, cada modelo emite una probabilidad para cada clase (por ejemplo, probabilidad de que la reseña sea positiva o negativa). Luego, se promedian estas probabilidades para obtener la predicción final. En este escenario, el soft voting es útil cuando se desea tener en cuenta la confianza o certeza de cada modelo, ya que se consideran las probabilidades en lugar de decisiones binarias.

4. ANÁLISIS DE DATOS

4.1 BASE DE DATOS

Para la investigación, se utilizaron los datos recopilados por el CIDEIM. Los archivos incluyen 27 fotografías de lesiones de pacientes, cada una etiquetada con su respectivo atributo objetivo: "falla" (no cura) o "cura". El 40.74% de dichas fotografías se define como "falla" y el 59.26% como "cura", lo que indica un buen balance en el conjunto de imágenes.

Además, se dispone de una base de datos de información transcriptómica que abarca 31 pacientes. El 38.71% de estos pacientes corresponde a "falla" y el 61.29% corresponde a "cura", indicando que las clases se encuentran balanceadas. Asimismo, se cuenta con 17,530 atributos correspondientes a genes, y el atributo objetivo que clasifica entre "falla" (no cura) o "cura". Cada gen está asociado con su valor CPM, que representa el número de lecturas específicas para dicho gen en el proceso de secuenciación.

La base de datos de transcriptomas proporciona detalles sobre los genes específicos de cada paciente. A continuación, se presenta un ejemplo de esta información, la columna “Gen” contiene los genes identificados en el paciente y la columna “CPM” hace referencia a Counts Per Million que representa el número de lecturas específicas para un gen en particular en el proceso de secuenciación:

Outcome	Cura
Gen	**CPM**
ENSG00000000003	4,705391664
ENSG00000000005	0
ENSG00000000419	31,62023198
ENSG00000000457	31,05558498
ENSG00000000460	11,66937133
⋮	⋮
ENSG00000288053	0
ENSG00000288436	0,376431333
ENSG00000288520	0,062738556
ENSG00000288534	0,250954222
ENSG00000288547	0,564647
ENSG00000288558	0,125477111

Tabla 1: ***Transcriptoma parcial del paciente su1154.***
Fuente: Bases de datos CIDEIM.

4.2 IDENTIFICACIÓN Y PREPARACIÓN DE DATOS:

Tras recibir la información de las fotografías de las lesiones de los pacientes y la base de datos de transcriptomas, se procedió a asignar un identificador único a cada paciente.

Dado que la base de datos del transcriptoma consta de 17,530 atributos (genes), el CIDEIM realizó un análisis preliminar que permitió identificar genes con diferencias significativas entre la primera y segunda visita de tratamiento de cada uno de los 31 pacientes. Emplearon la técnica del logaritmo en base 2 del fold change (cambio de expresión en visita 1 vs. visita 2), una medida común en el análisis de expresión génica. Esta técnica cuantifica las diferencias en la expresión de un gen entre dos situaciones o condiciones experimentales, facilitando la identificación de genes con una expresión más pronunciada en una visita en comparación con la siguiente. De este modo, se proporciona la razón de cambio de cada gen en el contraste evaluado [64].

En el análisis de expresión génica, se buscan genes cuya expresión haya aumentado o disminuido al menos 2 veces ($log2fc \geq 1$ o $log2fc \leq -1$), con un p-valor ≤ 0.05. El p-valor, indicador de la significancia estadística, sugiere que las diferencias observadas son estadísticamente relevantes. Este análisis se realiza mediante DESeq2, una herramienta bioinformática que utiliza modelos estadísticos para identificar genes con expresión diferencial en conjuntos de datos de expresión génica, teniendo en cuenta la variabilidad para obtener resultados más precisos [64].

Con base en esta información, la investigación buscó establecer un criterio que permitiera la selección de los genes más relevantes, con el objetivo de reducir la dimensionalidad del conjunto de datos.

4.3 PREPROCESAMIENTO DEL CONJUNTO DE DATOS DE TRANSCRIPTOMAS:

En la etapa inicial del preprocesamiento de los datos de transcriptomas, se reorganizaron las columnas mediante la transposición de la información. Esta acción facilitó la conversión de los ID de los pacientes y sus estados en registros, mientras que los genes se dispusieron como columnas. La reestructuración fue esencial ya que los genes originalmente estaban dispuestos de forma vertical y los pacientes de manera horizontal. La transposición de los datos permitió asociar los genes como atributos específicos de cada paciente. Posteriormente, este conjunto de datos se dividió en dos conjuntos: entrenamiento y prueba, representando el 70% y 30% del total del conjunto de datos, respectivamente.

Durante la limpieza de los datos, se excluyó el ID del paciente, ya que no influiría en los análisis subsiguientes. Únicamente se conservó el estado como variable objetivo: "cura" o "falla". Estos estados se codificaron asignando el número (1) a "falla" y el número (0) a "cura", estableciendo

así las "fallas" (no cura) como la clase objetivo de la investigación y los genes como las características relevantes.

Acorde a la estructura inicial del conjunto de los datos, se identificó una alta dimensionalidad debido a la gran cantidad de genes asociados a cada paciente. Por consiguiente, se eliminaron los genes que presentaban valores nulos, ya que en este contexto no era factible completar los campos vacíos debido a su posible impacto en las estimaciones. Tras la exclusión de estos genes, se seleccionaron los 100 genes con los cambios positivos más altos y los 100 genes con los cambios negativos más bajos, según el análisis previo proporcionado por el CIDEIM mediante el uso del DESeq2. Este procedimiento permitió reducir considerablemente la dimensionalidad de los atributos de interés, seleccionando un total de 171 genes de interés de los 31 pacientes de la base de datos.

La reducción inicial de la dimensionalidad logró disminuir significativamente los atributos en el conjunto de datos. A pesar de esta reducción, y debido al número limitado de pacientes, aún se conservaba un conjunto considerablemente grande de genes. En consecuencia, se optó por aplicar múltiples técnicas de reducción de la dimensionalidad, específicamente diseñadas para abordar un problema de clasificación binaria (cura o falla). Siguiendo la recomendación de expertos en el campo[1], el objetivo era limitar el resultado a un máximo de tres genes considerando la cantidad de pacientes disponibles para la investigación (31) para evitar así el sobreajuste (overfitting) y mantener un equilibrio adecuado entre la cantidad de datos y la complejidad del modelo.

La primera técnica empleada fue el Análisis de Varianza (ANOVA) para determinar si existían diferencias significativas entre las medias de los grupos. Se utilizó la librería scikit-learn en Python para inicializar un selector con el método ANOVA, estableciendo el número de características a seleccionar como k=3. De este modo, se identificaron y seleccionaron las características deseadas, mostrando los índices resultantes: [45, 46, 37], que corresponden a los genes: ENSG00000166211, ENSG00000198542 y ENSG00000177098.

La segunda técnica empleada para reducir la dimensionalidad de los 171 genes fue SVM-RFE (Recursive Feature Elimination based on Support Vector Machines), diseñada para seleccionar características en conjuntos de datos. En esta técnica, el SVM busca un hiperplano en un espacio de múltiples dimensiones para separar clases de datos, mientras que RFE elimina características

[1] Maria Adelaida Gomez, Coordinadora del Laboratorio de Bioquímica y Biología Molecular y Lina Fernanda Giraldo Parra, Bacterióloga y Laboralista Clínica, que hacen parte del Centro Internacional de Entrenamiento e Investigaciones Médicas (CIDEIM) y han realizado investigaciones sobre diferentes aspectos de la leishmaniasis.

menos relevantes recalibrando la precisión del modelo en cada iteración [22]. El SVM se entrenó con el conjunto de entrenamiento y se evaluó su rendimiento en el conjunto de prueba. Como resultado, se identificaron los genes más relevantes: ENSG00000157601, ENSG00000122862 y ENSG00000117594.

La tercera técnica aplicada para reducir la dimensionalidad en el conjunto de datos de 171 genes fue el RFE (Recursive Feature Elimination) utilizando Árboles de Decisión. Esta estrategia de selección de características combina la capacidad de los árboles de decisión con la eficacia de RFE para identificar las variables más significativas en un conjunto de datos [47]. Se entrenó el modelo de árboles de decisión con el conjunto de entrenamiento y luego se evaluó su rendimiento en el conjunto de prueba. Este procedimiento permitió alcanzar un conjunto de tres características, identificando los genes más relevantes como: ENSG00000198542, ENSG00000166736 y ENSG00000125144.

La cuarta técnica aplicada para reducir la dimensionalidad fue el Análisis de Componentes Principales (PCA Este método se implementó en el conjunto de entrenamiento después de estandarizar los datos. Inicialmente, se optó por trabajar con 10 componentes principales, los cuales abarcaron alrededor del 83% de la contribución relativa con respecto a la varianza total de los datos. Debido a la limitación de pacientes, se procedió a la selección de tres componentes principales, obteniendo los genes: ENSG00000155269, ENSG00000154451 y ENSG00000205358, representando una contribución relativa de aproximadamente el 44%.

A continuación, se detallan los genes seleccionados por cada una de las técnicas de reducción de dimensionalidad mencionadas:

TECNICA REDUCCIÓN DIMENSIONALIDAD	GENES SELECCIONADOS
ANOVA	[ENSG00000166211, ENSG00000198542, ENSG00000177098]
RFE con SVM	[ENSG00000157601, ENSG00000122862, ENSG00000117594]
RFE con árbol de decisión	[ENSG00000198542, ENSG00000166736, ENSG00000125144]
PCA	[ENSG00000155269, ENSG00000154451, ENSG00000205358]

Tabla 2: ***Genes resultantes del proceso de reducción de dimensionalidad ..***
Fuente: Elaboración propia a partir a partir de estimaciones con datos del CIDEIM.

4.4 PREPROCESAMIENTO DE CONJUNTO DE IMÁGENES DE LESIONES:

En el caso del conjunto de imágenes proporcionadas por el CIDEIM, se procedió a la asignación de etiquetas, clasificándolas según su estado de curación o falla (no cura). Asimismo, el

preprocesamiento de las imágenes de lesiones de leishmaniasis atravesó diversas etapas destinadas a mejorar la calidad y utilidad de los datos. La primera consistió en el recorte de imágenes, donde se implementó un procedimiento sistemático para eliminar el ruido y focalizarse en áreas específicas de las lesiones. Este enfoque posibilitó la eliminación de partes irrelevantes de las imágenes, concentrándose exclusivamente en las zonas de interés para el análisis de leishmaniasis, como se ilustra en la figura adjunta.

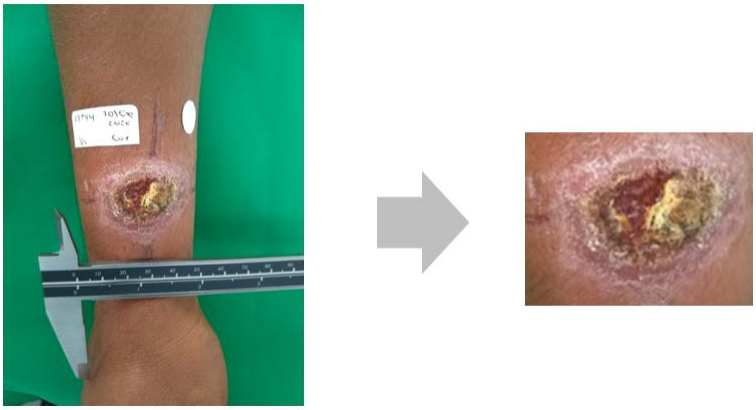


Ilustración 3: ***Lesión segmentada.***
Fuente: Elaboración propia a partir de datos del CIDEIM.

Una vez que se cargaron las imágenes desde un directorio especificado, se ajustaron a las dimensiones predeterminadas (224x224 píxeles) para redimensionarlas. Luego, se convirtieron las imágenes en matrices y las etiquetas correspondientes se almacenaron en un arreglo.

Posteriormente, el conjunto de datos se dividió en una proporción del 80/20, asignando el 80% de las imágenes al conjunto de entrenamiento y reservando el 20% restante para el conjunto de pruebas. Esta división aseguró una segmentación adecuada para entrenar utilizando validación cruzada y evaluar el modelo de manera independiente.

Dado la poca cantidad de fotografías de lesiones, se aplicó un aumento de datos utilizando diversas técnicas. Estas permitieron generar hasta siete imágenes a partir de una única imagen, lo cual fue esencial para mejorar el rendimiento de los modelos.

La primera técnica de aumentación de datos aplicada fue la rotación, que consistió en realizar transformaciones de espejo tanto horizontal como vertical en las imágenes. Este proceso ayudó a capturar variaciones adicionales.

Como segunda técnica, se empleó el ajuste de brillo, ejecutando múltiples variaciones en la

iluminación. Sin embargo, se seleccionaron únicamente aquellas variaciones que no afectaron la calidad de la imagen de la lesión, descartando aquellos en los cuales no se observaba la lesión correctamente como se muestra a continuación:

Ilustración 4: ***Depuración de cambios en el brillo***
Fuente: Elaboración propia a partir de datos del CIDEIM.

La última técnica aplicada consistió en la rotación en ángulos de 90, 180 y 270 grados. Se procesaron las imágenes originales en los grados mencionados con el objetivo de introducir variabilidad en la orientación de las lesiones.

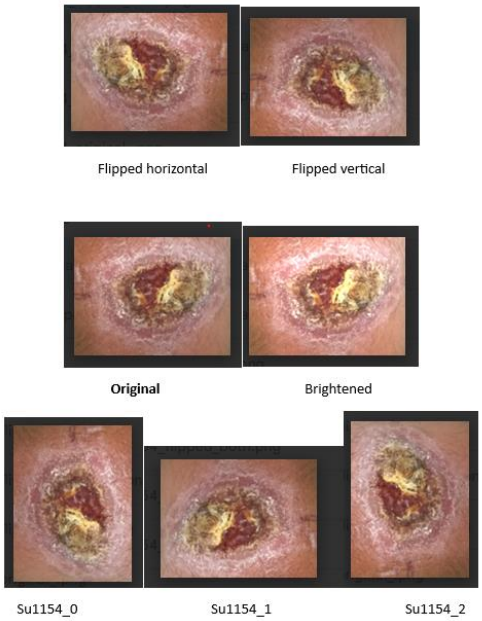


Ilustración 5: ***Resultado del aumento de datos***

Fuente: Elaboración propia a partir de datos del CIDEIM.

Después de aplicar estas transformaciones, se generaron nuevas instancias de imágenes enriquecidas; estas tres técnicas se aplicaron al set de imágenes de entrenamiento y como se mencionó anteriormente con la limitación de imágenes originales se aplicó únicamente la técnica de rotación en el conjunto de prueba con el fin de realizar las mejores estimaciones. Para cada imagen según su estatus de cura y falla (no cura) se asignó una etiqueta 1 para falla y 0 para cura, de esta manera se obtuvieron 168 imágenes en el conjunto de entrenamiento y 24 imágenes en el conjunto de prueba.

Posterior realizamos una conversión del tipo de dato de las matrices a flotante de 32 bits, esta conversión era beneficiosa en términos de eficiencia computacional y uso de memoria ya que los modelos a aplicar correspondían al aprendizaje profundo y por lo tanto se necesitaba optimizar el rendimiento y se requería que los datos de entrada estuviesen en este formato. Una vez realizado este proceso se normalizaron los datos dividiendo los valores de los píxeles de las imágenes por 255.0, escalándolos al rango de 0 a 1 para facilitar la convergencia durante el entrenamiento.

5. MODELADO

5.1 MODELOS DE APRENDIZAJE AUTOMATICO:

5.1.1 Construcción de modelos base:

Inicialmente, se construyeron una serie de modelos base, que fueron entrenados con el conjunto de entrenamiento y evaluados con el conjunto de prueba. Estos modelos base se crearon utilizando las funciones proporcionadas por scikit-learn, una biblioteca de aprendizaje automático en Python que ofrece herramientas eficientes para la implementación de diversos algoritmos y técnicas de modelado.

El primer modelo base aplicado fue el SVC (Support Vector Classifier):

$$SVC = SVC(kernel =' linear')$$

El segundo modelo base planteado fue un árbol de decisión, partiendo de un modelo básico para clasificación:

$$DT = DecisionTreeClassifier()$$

El tercer modelo base fue el k-vecinos, implementado con la estructura básica para clasificación:

$$KNN = KNeighborsClassifier()$$

Finalmente, el último modelo base construido fue el de Bosques Aleatorios, implementado en su formato más simple:

$$RF = RandomForestClassifier(random_{state} = 42)$$

5.1.2 Validación y evaluación de modelos base:

Se realizaron las predicciones sobre el conjunto de prueba y se evaluó el rendimiento de cada modelo de clasificación. Se utilizaron las funciones "accuracy_score" y "f1_score" del módulo "sklearn.metrics". La precisión, que representa la fracción de predicciones correctas en relación con las etiquetas verdaderas, y el puntaje F1, que considera tanto la precisión como la exhaustividad del modelo, fueron las métricas evaluadas.

A continuación, se presentan los resultados obtenidos para cada modelo con el conjunto de datos de prueba, indicando los genes resultantes de cada técnica utilizada:

ANOVA- MODELO BASE								
	SVM		Árbol decisión		K-vecinos		Bosque aleatorio	
GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ENSG00000166211 ENSG00000198542 ENSG00000177098	0,8	0,75	0,7	0,57	**0,9**	**0,86**	0,8	0,75

*Tabla 3: 1**Resultados de los modelos con genes seleccionados mediante ANOVA.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

RFE - MODELO BASE								
		SVM		Árbol decisión		K-vecinos		Bosque aleatorio	
Método	GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	0,7	0,57	0,8	0,66	0,7	0,57	0,8	0,67
RFE ARBOLES	ENSG00000198542 ENSG00000166736 ENSG00000125144	**0,8**	**0,75**	**0,8**	**0,75**	0,5	0,44	**0,8**	**0,75**

*Tabla 4: **Resultados de los modelos con genes seleccionados mediante RFE.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

PCA - MODELO BASE								
	SVM		Arboles		K-vecinos		Bosque aleatorio	
GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ENSG00000155269 ENSG00000154451 ENSG00000205358	0,8	0,5	0,8	0,75	**0,9**	**0,8**	**0,9**	**0,86**

*Tabla 5: **Resultados de los modelos con genes seleccionados mediante PCA.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Los genes seleccionados a través de PCA mostraron los mejores resultados en la mayoría de los modelos, según los datos recopilados. En contraste, los genes de las técnicas RFE y ANOVA mostraron resultados bastante similares entre sí. Estos hallazgos proporcionan una primera aproximación para realizar estimaciones que pudieran mejorar los resultados obtenidos.

5.1.3 Optimización de los modelos base:

En este punto, con los corpus de entrenamiento se llevó a cabo la validación cruzada junto con la búsqueda en cuadrícula para encontrar la combinación óptima de hiperparámetros que maximizara la precisión de cada modelo. En el contexto de la validación cruzada, no se utilizó un conjunto de validación separado de manera independiente. En cambio, se realizaron múltiples divisiones del conjunto de entrenamiento en conjuntos de entrenamiento para ajustar y validar

los modelos: Máquinas de Soporte Vectorial (SVM), Árboles de Decisión, k-Vecinos Más Cercanos (KNN) y Bosques Aleatorios; evitando así el riesgo de sobreajuste y permitiendo una configuración de modelo más generalizable a datos no vistos.

Luego de obtener los mejores hiperparámetros, se procedió a evaluar los modelos finales en el conjunto de prueba. Esta evaluación final proporcionó una medida realista del rendimiento de los modelos en datos no utilizados durante el entrenamiento y ajuste de hiperparámetros, validando así la capacidad de generalización de cada modelo.

Se utilizó la metodología de grilla, la cual facilita la exploración de diversas combinaciones de hiperparámetros con el objetivo de identificar aquellos que maximizasen el rendimiento del modelo en el conjunto de datos. En la primera fase, se identificaron los hiperparámetros clave para cada modelo. Posteriormente, se construyó una grilla que abarcara todas las combinaciones de valores potenciales para los hiperparámetros del modelo de interés.

Para el modelo SVM de clasificación, se definió un espacio de búsqueda para los hiperparámetros, considerando el tipo de '$Kernel$', el parámetro de regulación 'c ', el parámetro '$shrinking$' para determinar si se debe utilizar o no la heurística de contracción para acelerar el proceso de entrenamiento, la '$probability$' que indica si se deben calcular las probabilidades para las clases y el '$\max\_iter$' para establecer el número máximo de iteraciones permitidas durante la optimización.

Para el modelo de k-vecinos, se elaboró un espacio de búsqueda para los hiperparámetros. El primero de ellos fue el parámetro '$neighbors$' para determinar el número de vecinos utilizado en la clasificación. Luego, se consideró el parámetro '$weights$' para identificar el tipo de peso asignado a los vecinos, ya sea que todos tengan el mismo peso o que los más cercanos tengan un peso mayor. El último parámetro fue 'p', que define la métrica de distancia utilizada.

En el caso de los árboles de decisión, el espacio de búsqueda de los hiperparámetros abarcó el '$\max\_depth$', el cual establece la profundidad máxima del árbol, controlando la longitud máxima de cualquier camino desde la raíz hasta una hoja. También se consideraron el '$\min\_samples\ split$' para establecer el número mínimo de muestras requeridas para dividir un nodo interno junto con el parámetro '$\min\_samples\ leaf$' para definir el número mínimo de muestras requeridas para que un nodo sea considerado una hoja, Además, se incluyó el '$\max\_leaf\ nodes$' que define el máximo de nodos hoja permitidos en el árbol, el '$\min\_impurity\ decrease$' que puede ayudar a controlar la complejidad del modelo y finalmente el '$criterion$' para medir la calidad de una división.

Finalmente, para los árboles aleatorios, la estimación de hiperparámetros se enfocó en los '$n\ estimators$ 'para establecer el número de árboles en el bosque, y se tuvieron en cuenta algunos de los hiperparámetros establecidos para los árboles de decisión como lo son el 'max $\_\ depth$', el 'min $\_samples\ split$'y el 'min $\_\ samples\ leaf$'.

En este proceso, enfrentamos un desafío significativo al buscar los hiperparámetros óptimos. Se crearon diversos conjuntos de valores que se probaron y compararon con los resultados del modelo base, descartando y ajustando sucesivamente hasta alcanzar los mejores resultados.

Posteriormente, se procedió a entrenar los modelos para cada combinación, utilizando el conjunto de entrenamiento y aplicando validación cruzada. Finalmente, se evaluaron con el conjunto de datos de prueba, seleccionando las mejores combinaciones de hiperparámetros basándonos en la métrica de exactitud.

A continuación, se detallan los rangos de búsqueda de los hiperparámetros y los mejores hiperparámetros obtenidos para cada uno de los conjuntos de datos definidos previamente mediante las técnicas de reducción de dimensionalidad:

Método Reducción dimensionalidad	Genes	Modelo	Hiperparámetros	Rango Hiperparámetros	Mejores Hiperparámetros
ANOVA	ENSG00000166211 ENSG00000198542 ENSG00000177098	SVM	kernel	['linear', 'poly', 'rbf', 'sigmoid']	**linear**
			C	[0.12, 0.13]	**0.12**
			shrinking	[True, False]	**True**
			probability	[True, False]	**True**
			class_weight	[None, 'balanced']	**None**
			max_iter	[2, 5, 10, 20, 30]	**10**
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	SVM	kernel	['linear', 'poly', 'rbf', 'sigmoid']	**poly**
			C	[0.1, 1, 10]	**1**
			shrinking	[True, False]	**True**
			probability	[True, False]	**True**
			class_weight	[None, 'balanced']	**None**
			max_iter	[800, 1000, 1200, 1300]	**800**
RFE DT	ENSG00000198542 ENSG00000166736 ENSG00000125144	SVM	kernel	['linear', 'poly', 'rbf', 'sigmoid']	**linear**
			C	[0.3, 0.6, 1, 10]	**0.3**
			shrinking	[True, False]	**True**
			probability	[True, False]	**True**
			class_weight	[None, 'balanced']	**None**
			max_iter	[1000, 1500, 2000, 2500]	**1000**
PCA	ENSG00000155269 ENSG00000154451	SVM	kernel	['linear', 'poly', 'rbf', 'sigmoid']	**poly**
			C	[0.1, 1, 5, 10]	**10**

	ENSG00000205358		shrinking	[True, False]	**False**
			probability	[True, False]	**True**
			class_weight	[None, 'balanced']	**balanced**
			max_iter	[20, 50, 100, 200, 400]	**50**

Tabla 6: ***Resultados de la Optimización de Hiperparámetros para SVM.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Método Reducción dimensionalidad	Genes	Modelo	Hiperparámetros	Rango Hiperparámetros	Mejores Hiperparámetros
ANOVA	ENSG00000166211 ENSG00000198542 ENSG00000177098	K-Vecinos	n_neighbors	[3, 5, 7, 9]	**7**
			weights	['uniform', 'distance']	**uniform**
			p	[1, 2]	**1**
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	K-Vecinos	n_neighbors	[3, 5, 7, 9]	**5**
			weights	['uniform', 'distance']	**uniform**
			p	[1, 2]	**2**
RFE DT	ENSG00000198542 ENSG00000166736 ENSG00000125144	K-Vecinos	n_neighbors	[3, 5, 7, 9]	**3**
			weights	['uniform', 'distance']	**distance**
			p	[1, 2]	**2**
PCA	ENSG00000155269 ENSG00000154451 ENSG00000205358	K-Vecinos	n_neighbors	[3, 5, 7, 9]	**7**
			weights	['uniform', 'distance']	**uniform**
			p	[1, 2]	**2**

Tabla 7: ***Resultados de la Optimización de Hiperparámetros para K-vecinos.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Método Reducción dimensionalidad	Genes	Modelo	Hiperparámetros	Rango Hiperparámetros	Mejores Hiperparámetros
ANOVA	ENSG00000166211 ENSG00000198542 ENSG00000177098	Árbol decisión	max_depth	[None, 5, 10, 15, 20]	**None**
			min_samples_split	[3, 5, 10, 15]	**10**
			min_samples_leaf	[2, 4, 8, 10]	**8**
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	Árbol decisión	max_depth	[None, 5, 10, 15, 20]	**None**
			min_samples_split	[2, 3, 4, 5, 10]	**2**
			min_samples_leaf	[2, 4, 5, 6, 8]	**4**
			criterion	['gini', 'entropy']	**gini**
			max_leaf_nodes	[None, 2, 5, 10]	**None**
			min_impurity_decrease	[0.0, 0.1, 0.2]	**0.0**
RFE DT	ENSG00000198542 ENSG00000166736 ENSG00000125144	Árbol decisión	max_depth	[None, 5, 10, 15, 20]	**None**
			min_samples_split	[2, 3, 4, 5, 10]	**10**
			min_samples_leaf	[2, 4, 5, 6, 8]	**2**
			criterion	['gini', 'entropy']	**gini**

			max_leaf_nodes	[None, 2, 5, 10]	**None**
			min_impurity_decrease	[0.0, 0.1, 0.2]	**0.0**
PCA	ENSG00000155269 ENSG00000154451 ENSG00000205358	Árbol decisión	max_depth	[None, 5, 10, 15, 20]	**None**
			min_samples_split	[2, 3, 4, 5, 10]	**2**
			min_samples_leaf	[2, 4, 5, 6, 8]	**6**
			criterion	['gini', 'entropy']	**gini**
			max_leaf_nodes	[None, 2, 5, 10]	**None**
			min_impurity_decrease	[0.0, 0.1, 0.2]	**0.0**

Tabla 8: ***Resultados de la Optimización de Hiperparámetros para Árbol de decisión.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Método Reducción dimensionalidad	Genes	Modelo	Hiperparámetros	Rango Hiperparámetros	Mejores Hiperparámetros
ANOVA	ENSG00000166211 ENSG00000198542 ENSG00000177098	Bosque Aleatorio	n_estimators	[10, 20, 50, 100, 200]	**10**
			max_depth	[None, 10, 20, 30]	**None**
			min_samples_split	[1, 2, 5, 10]	**2**
			min_samples_leaf	[1, 2, 5, 6]	**1**
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	Bosque Aleatorio	n_estimators	[50, 60, 75, 70]	**50**
			max_depth	[None, 1, 2, 5, 10, 20]	**None**
			min_samples_split	[3, 5, 15, 20]	**3**
			min_samples_leaf	[1, 2, 4, 5, 6]	**4**
			max_leaf_nodes	[None, 10, 20, 30]	**None**
RFE DT	ENSG00000198542 ENSG00000166736 ENSG00000125144	Bosque Aleatorio	n_estimators	[50, 60, 75, 70]	**75**
			max_depth	[None, 1, 2, 5, 10, 20]	**None**
			min_samples_split	[3, 5, 15, 20]	**3**
			min_samples_leaf	[1, 2, 4, 5, 6]	**2**
			max_leaf_nodes	[None, 10, 20, 30]	**None**
PCA	ENSG00000155269 ENSG00000154451 ENSG00000205358	Bosque Aleatorio	n_estimators	[50, 60, 75, 70, 80]	**70**
			max_depth	[None, 1, 2, 5, 10, 20]	**None**
			min_samples_split	[2, 5, 10, 15, 20]	**2**
			min_samples_leaf	[1, 2, 4]	**4**

Tabla 9: ***Resultados de la Optimización de Hiperparámetros para Bosque Aleatorio.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

5.1.4 Validación y evaluación de los modelos optimizados:

Después de identificar los hiperparámetros óptimos, cada modelo se entrenó con el conjunto de datos de entrenamiento utilizando esos parámetros seleccionados. Luego, se llevó a cabo la validación de los modelos con el conjunto de datos de prueba utilizando las mismas métricas

que en los modelos base: "accuracy_score" y "f1_score" del módulo "sklearn.metrics".

Los resultados obtenidos tras la ejecución de los modelos son los siguientes:

ANOVA - MODELO ESTIMADO								
	SVM		Arboles		K-vecinos		Bosque Aleatorio	
GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ENSG00000166211 ENSG00000198542 ENSG00000177098	0,9	0,85	0,8	0,75	0,9	0,86	0,8	0,75

Tabla 10: ***Resultados de los modelos optimizados con los genes identificados a través de ANOVA.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

RFE - MODELO ESTIMADO								
		SVM		Arboles		K-vecinos		Bosque Aleatorio	
Método	GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RFE SVM	ENSG00000157601 ENSG00000122862 ENSG00000117594	0,9	0,8	0,8	0,66	0,7	0,57	0,7	0,57
RFE ARBOLES	ENSG00000198542 ENSG00000166736 ENSG00000125144	0,9	0,85	0,8	0,75	0,4	0,4	0,8	0,75

Tabla 11: ***Resultados de los modelos optimizados con los genes identificados a través de RFE.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

PCA - MODELO ESTIMADO								
	SVM		Arboles		K-vecinos		Bosque Aleatorio	
GEN	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ENSG00000155269 ENSG00000154451 ENSG00000205358	1	1	1	1	1	1	1	1

Tabla 12: ***Resultados de los modelos optimizados con los genes identificados a través de PCA.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Con base en los resultados obtenidos, se observa que el SVM experimentó mejoras significativas en casi todas las técnicas tras la estimación de los hiperparámetros. En contraste, otros modelos como los árboles de decisión o k-vecinos no mostraron variaciones sustanciales con los mejores hiperparámetros. En relación con los resultados obtenidos para PCA, se evidencia un Accuracy y un F1 iguales a uno. Esto sugiere la necesidad de un conjunto de datos más extenso para realizar pruebas adicionales y analizar en detalle el comportamiento de este modelo. Sin embargo, dada la limitación de información disponible en el momento de esta investigación, este comentario se deja como una sugerencia para futuras investigaciones.

5.2 REDES CONVOLUCIONALES:

5.2.1 Construcción modelo red neuronal base:

Se llevó a cabo una validación cruzada estratificada de 5 pliegues en una red neuronal convolucional (CNN) mediante Keras. Se utilizó la técnica StratifiedKFold para asegurar una distribución equitativa de clases en cada pliegue. Además, se configuró el Early Stopping para monitorear la pérdida en el conjunto de validación y detener el entrenamiento si la pérdida no mejora después de 5 épocas. Posteriormente, se definió un conjunto de entrenamiento y prueba a partir del conjunto de datos de entrenamiento inicial. Cabe destacar que la validación cruzada se empleó como una técnica de entrenamiento para evaluar el rendimiento del modelo en diferentes subconjuntos de datos, garantizando así su robustez y generalización.

El modelo definido fue una red neuronal convolucional (CNN) con capas específicas. Comenzando con una capa Conv2D que utilizó 16 filtros de 3x3 para extraer características de imágenes de entrada con forma (224, 224, 3). La función de activación ReLU se aplicó en esta capa para introducir no linealidades.

Después de la capa de convolución, se agregó una capa Flatten para transformar los datos a un formato unidimensional. Luego, se incorporaron dos capas Dense. La primera capa densa tenía 64 neuronas con activación ReLU, y la segunda capa densa tenía 1 neurona con activación sigmoide, indicando que el modelo se utilizó para la clasificación binaria (0 o 1).

El modelo se compiló utilizando el optimizador Adam, una elección común para problemas de aprendizaje profundo. La función de pérdida seleccionada fue la entropía cruzada binaria, apropiada para problemas de clasificación binaria. Además, se monitoreó la precisión durante el entrenamiento y la evaluación como métrica de rendimiento.

En resumen, el modelo combinó capas convolucionales y densas para aprender representaciones de características y realizar la clasificación. Se configuró para optimizar mediante el algoritmo Adam, minimizando la entropía cruzada binaria, mientras se evaluaba la precisión durante el proceso de entrenamiento y evaluación.

5.2.2 Validación y evaluación de modelo red neuronal base:

Se evaluó el modelo con el conjunto de prueba para las métricas de precisión media con su desviación estándar y la pérdida media de todos los pliegues. Esto proporcionó una evaluación más robusta del rendimiento del modelo en comparación con una única ejecución.

$$Accuracy: 77.40\%(+/-9.56\%)$$

$Loss: 0.49$

La precisión promedio del modelo en los 5 pliegues es del 77.40%, con una desviación estándar de ±9.56% lo que indica la variabilidad de la precisión entre los pliegues. Esta desviación estándar es relativamente alta y puede ser un indicativo que el rendimiento del modelo varía significativamente entre diferentes divisiones del conjunto de datos. Y por otro lado la pérdida promedio del modelo en los 5 pliegues es de 0.49 que corresponde a una medida de cuán bien el modelo está aprendiendo. En problemas de clasificación binaria, como este, se mide mediante la entropía cruzada binaria (binary crossentropy). Con los resultados obtenidos se construyó un gráfico de la función de pérdida (loss) del modelo en función de las épocas durante el entrenamiento.

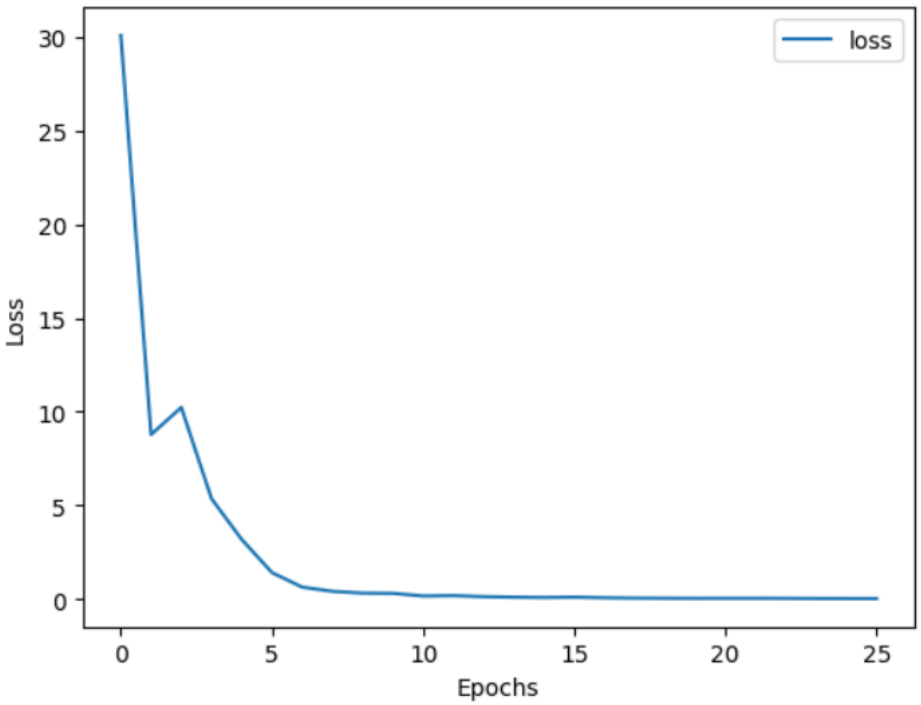


Ilustración 6: ***Función de pérdida para conjunto de datos de entrenamiento por cada época.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

5.2.3 Optimización del modelo red neuronal base:

Con el modelo base se inició la búsqueda de los mejores hiperparámetros utilizando Keras Tuner. Igualmente aplicando validación cruzada estratificada con 5 pliegues y el Early Stopping, se definió una función $build\ model$ que toma un objeto hp (hiperparametros) como argumento y se construyó la red con hiperparámetros ajustables. Para la estimación se tuvieron en cuenta el

número de filtros, el tamaño del kernel, la función de activación, el número de unidades en la capa densa, la función de activación en la capa densa y la tasa de aprendizaje.

Capas	Parámetros	Valores	Mejor valor
Capa convolucional	**Filtros**	12,14,16,24	**16**
	kernel	1,2,3	**3**
	Activación	['relu', 'tanh', 'elu']	**Elu**
Capa densa	**Unidades**	24, 32,64	**64**
	Activación	['relu', 'tanh', 'elu']	**Tanh**

Tabla 13: ***Resultados de la estimación de hiperparametros del modelo base.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Se estableció el número máximo de pruebas $(\max t\,rials)$ en 20, probando hasta 20 combinaciones diferentes de hiperparámetros. Después de la búsqueda, se obtuvo el siguiente modelo.

```
Mejor modelo:
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 222, 222, 16)      448

 flatten_1 (Flatten)         (None, 788544)            0

 dense_2 (Dense)             (None, 64)                50466880

 dense_3 (Dense)             (None, 1)                 65

=================================================================
```

Ilustración 7: ***Arquitectura Red convolucional base.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

5.2.4 Validación y evaluación del modelo red neuronal base optimizado:

Se entrenó la red neuronal convolucional con el conjunto de entrenamiento utilizando los

mejores hiperparámetros encontrados durante la búsqueda y luego se evaluó su rendimiento en cada pliegue de la validación cruzada, se entrenó utilizando los datos de entrenamiento durante un máximo de 100 épocas utilizando "Detención temprana" con el fin de detener el entrenamiento si la pérdida en el conjunto de validación no mejoraba después de 5 épocas y posterior se evaluó en el conjunto de prueba como el modelo base.

Como en el modelo base se obtuvo la precisión media con su desviación estándar y la pérdida media de todos los pliegues. Obteniendo los siguientes resultados:

$$Accuracy: 78.62\%(+/-11.97\%)$$

$$Loss: 0.45$$

En comparación con los resultados del primer modelo (77.40%) se tuvo una precisión promedio más alta (78.62%). La desviación estándar fue ligeramente más alta (±11.97%) al modelo base (±9.56%) indicando una mayor variabilidad en el rendimiento entre pliegues. También se tuvo una pérdida promedio más baja (0.45) que indica un rendimiento ligeramente mejor de la red.

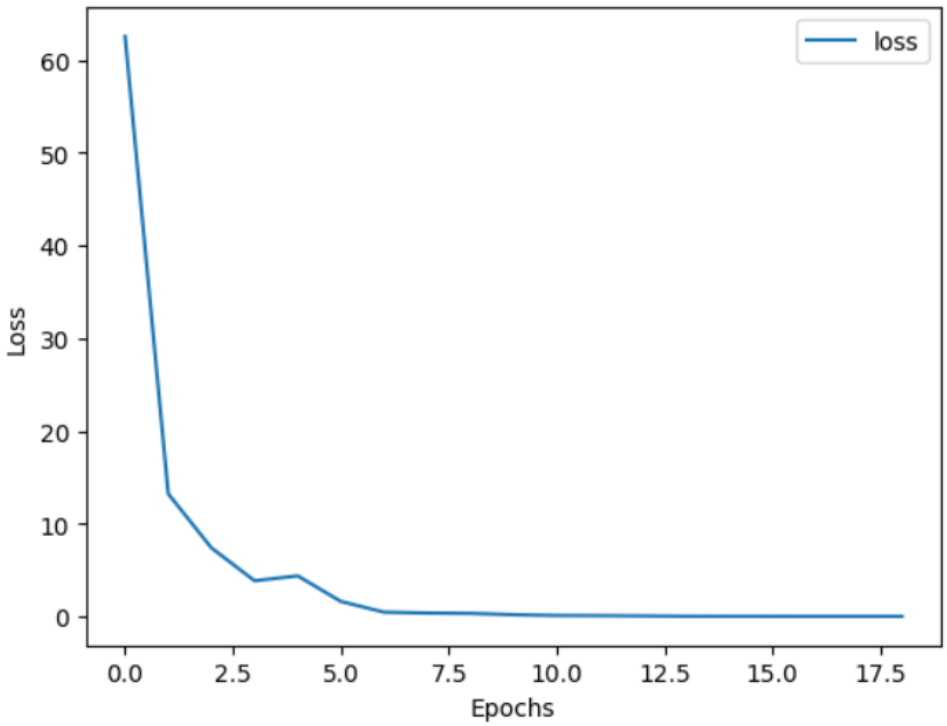


Ilustración 8: ***Función de pérdida para conjunto de datos de entrenamiento por cada época.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

5.2.5 Construcción de la red neuronal convolucional con arquitectura vgg16:

Para esta implementación, se recurrió a referencias y mejores prácticas proporcionadas por la documentación oficial de librerías como TensorFlow, Keras o PyTorch, así como a conocimientos adquiridos en la experimentación con técnicas de validación y entrenamiento de modelos de redes neuronales convolucionales.

En su implementación de transferencia de aprendizaje con la arquitectura VGG16, se estableció la capa de entrada para las imágenes en un formato específico de 224x224 píxeles con tres canales de color (224, 224,3). Esta configuración se ajustó para alinearse con las dimensiones requeridas para la arquitectura.

Además de definir la capa de entrada, se agregó una capa de aplanado (flatten) que transformó los mapas de características en un vector unidimensional, preparando así la información para la capa de salida. La capa de salida se diseñó con una sola neurona y función de activación sigmoide, especialmente indicada para problemas de clasificación binaria.

Para el proceso de entrenamiento, se optó por utilizar validación cruzada con k=5, dividiendo el conjunto de entrenamiento en cinco pliegues para entrenar el modelo. Además, se implementó early stopping en caso de no identificar una mejora después de 5 épocas, lo que ayudó a evitar el sobreajuste y acelera el entrenamiento.

Esta estrategia de entrenamiento con validación cruzada y early stopping permitió un ajuste más preciso del modelo y una mejor estimación del rendimiento en datos no vistos, contribuyendo así a la mejora del desempeño del modelo de clasificación con la arquitectura VGG16.

5.2.6 Validación y evaluación de la red neuronal convolucional con arquitectura vgg16:

Para el modelo Red Neuronal VGG19 se obtuvo la precisión media con su desviación estándar y la pérdida media de todos los pliegues. Obteniendo los siguientes resultados:

$$Accuracy: 91.84\%(+/-3.50\%)$$

$$Loss: 0.19$$

Con un accurancy promedio del 91.84%, con un rango de variación de aproximadamente +/- 3.50%, podemos considerar una consistencia robusta en el rendimiento del modelo a lo largo de diferentes pruebas y conjuntos de datos. Además, el valor de pérdida registrado fue de 0.19, indicando una buena capacidad del modelo para minimizar los errores durante el proceso de entrenamiento y validación. Estos resultados demuestran la efectividad y la capacidad predictiva destacada de la red neuronal implementada, respaldando su potencial para aplicaciones de

clasificación y reconocimiento con un alto nivel de confianza y precisión.

5.2.7 Optimización de la red neuronal convolucional con arquitectura vgg16:

En esta parte de optimización se procedió a crear otra red utilizando la arquitectura VGG16, la cual incluyó la capa de aplanado (flatten), una capa densa con un rango de unidades entre 32 y 256 y diversas funciones de activación como 'relu', 'elu' y 'tanh'. Esta red se configuró con una capa de salida con una unidad y función de activación sigmoide, adecuada para problemas de clasificación binaria.

5.2.8 Validación y evaluación de la red neuronal convolucional optimizada con arquitectura vgg16:

Durante este proceso, se aplicó validación cruzada para evaluar diferentes combinaciones de hiperparámetros y determinar los mejores valores para el modelo. Los resultados obtenidos fueron una precisión del 93.20% con una desviación estándar de +/- 3.78%, y una pérdida de 0.16.

$$Accuracy: 93.20\%(+/-3.78\%)$$

$$Loss: 0.16$$

Es importante destacar que este modelo se construyó utilizando validación cruzada con el conjunto de entrenamiento, lo que permitió evaluar su rendimiento en múltiples subdivisiones de los datos para obtener una estimación más robusta del desempeño del modelo.

Además, se seleccionó este modelo basado en los hiperparámetros que demostraron ser los más efectivos durante la validación cruzada. Esta combinación específica de configuración de red y parámetros resultó en un rendimiento sólido en la tarea de clasificación, con una alta precisión y una pérdida mínima.

5.2.9 Construcción de la red neuronal convolucional con arquitectura vgg19:

Durante la aplicación de transferencia de aprendizaje con la arquitectura VGG19 para la clasificación de casos entre "cura" o "falla", se siguió un enfoque específico. Inicialmente, se definió la capa de entrada para las imágenes en un tamaño de 224x224 píxeles con tres canales RGB. Luego, se agregó una capa de aplanado (flatten) y se configuró una capa de salida con una sola neurona y función de activación sigmoidal para lograr la clasificación binaria.

5.2.10 Validación y evaluación de la red neuronal convolucional con arquitectura vgg19:

El proceso de entrenamiento se llevó a cabo empleando con el conjunto de entrenamiento

validación cruzada con k=5 y se aplicó Detención temprana para prevenir el sobreajuste del modelo.

$$Accuracy: 91.15\%(+/-4.06\%)$$

$$Loss: 0.23$$

Los resultados obtenidos mostraron un promedio de precisión del 91.15%, con una desviación estándar del +/- 4.06%. Asimismo, la pérdida promedio registrada fue de 0.23, indicando una eficacia razonable en la reducción de errores durante el proceso de clasificación.

Estos resultados demuestran la capacidad del modelo para discernir entre las categorías "cura" y "falla" en las imágenes, con una consistencia razonable en su desempeño a lo largo de distintos conjuntos de validación. El enfoque de transferencia de aprendizaje utilizando la arquitectura VGG19 se presenta como una estrategia robusta y efectiva para la clasificación de imágenes en este contexto particular de diagnóstico médico.

5.2.11 Optimización de la red neuronal convolucional con arquitectura vgg19:

Posteriormente, se desarrolló una segunda red neuronal utilizando la arquitectura VGG19 con ciertas modificaciones. Esta red incluyó una capa de aplanado (flatten), seguida por una capa densa que varió el número de unidades entre 32 y 256, y se evaluaron diversas funciones de activación como 'relu', 'sigmoid', 'tanh'. Además, se exploraron diferentes valores de tasa de aprendizaje ('learning_rate') en el rango de [0.0001, 0.001]. Finalmente, se incorporó una capa de salida con una neurona y función de activación sigmoide, acorde con la naturaleza de la clasificación binaria.

5.2.12 Validación y evaluación de la red neuronal convolucional con arquitectura vgg19 optimizada:

Los resultados obtenidos en esta configuración mostraron una precisión promedio del 78.05%, con una desviación estándar del +/- 17.60%. Además, se registró una pérdida promedio de 0.41 durante el proceso de evaluación.

$$Accuracy: 78.05\%(+/-17.60\%)$$

$$Loss: 0.41$$

Estos resultados indican que, a pesar de haber obtenido una precisión menor en comparación con el modelo anterior, la red neuronal logró clasificar las imágenes con una precisión aceptable, aunque mostrando una mayor variabilidad en su rendimiento entre diferentes conjuntos de

validación. Esta nueva configuración explorada permitió una adaptación más flexible de la red, pero con una precisión promedio más baja y una mayor variabilidad en comparación con la primera red neuronal construida.

6. INTEGRACIÓN DE LOS MODELOS

Para realizar la integración de los modelos, el primer paso fue seleccionar cuales modelos se iban a tomar para la integración y cual conjunto de datos sería el elegido para dicho proceso. Lo primero fue revisar nuevamente los resultados obtenidos de los modelos supervisados: SVM, K-Vecinos, Arboles de Decisión y Bosque Aleatorio. En este paso no solo teníamos que seleccionar el mejor modelo sino también el que tuvo mejor comportamiento con los genes seleccionados aplicando las técnicas de reducción de dimensionalidad (ANOVA, PCA y RFE). Teniendo todos estos factores sobre la mesa se identificó que a pesar de que algunos de ellos tenían un comportamiento parecido los genes seleccionados por la técnica de ANOVA presentaron una mejora considerable en el modelo con los mejores hiperparámetros. Por lo cual se seleccionaron tres de los cuatros modelos probados para tener un número impar que permitiera construir el ensamble de una forma sencilla y completa. Los modelos seleccionados fueros SVM, K-Vecinos y Bosque Aleatorio.

Por otro lado, para el caso de las redes neuronales convoluciones se seleccionaron la red neuronal convolucional con la arquitectura VGG16 después de estimar los mejores hiperparámetros y la red neuronal convolucional con la arquitectura VGG19 básica; para la integración con los modelos supervisados mencionados anteriormente. Dichas redes fueron seleccionadas debido a que presentaron los mejores resultados dentro del modelado, y con esto completamos un total de cinco modelos para realizar la integración y aplicar el método de ensamble por votación ya que al tener un número impar la decisión final sería para la clase que mayor cantidad de votos tuviese.

Como lo hemos mencionado anteriormente, el conjunto de datos no es amplio y por lo tanto requeríamos seleccionar algunos casos para ejecutar este proceso. Para esto, se evaluaron uno a uno los IDs de los pacientes que conformaban el conjunto de datos de prueba tanto de los modelos SVM, K-Vecinos y Bosque Aleatorio como de las redes y de esta manera logramos identificar dos pacientes que se compartían en este conjunto. Uno de estos pacientes estaba clasificado como cura y el otro paciente como falla en el conjunto de datos original del CIDEIM. Una vez seleccionados con la información individual, se construyeron dos conjuntos de datos nuevos que contenían por un lado los transcriptomas y por el otro lado las imágenes de las

lesiones.

Estos conjuntos de datos se cargaron nuevamente, y se realizó las transformaciones necesarias. Para el caso algunos modelos de redes neuronales, se definió un umbral del 0.5 para convertir las predicciones probabilísticas en predicciones binarias. Una vez se tuvieron los datos y las imágenes finales, se realizó una votación por mayoría (hard voting) para combinar las predicciones de los modelos y obtener una predicción final, la cual se obtiene al dividir la suma de las predicciones por el número de modelos. En este caso, teníamos cinco modelos (preds_model_1, preds_model_2, preds_model_3, preds_red_1, y preds_red_2), por lo que se dividió por 5.

Por último, se utilizó la función $np.round$ para redondear la predicción final a 0 o 1. La elección de este enfoque permitió mejorar la robustez y la precisión del modelo final al aprovechar las fortalezas de cada modelo individual.

Se evaluaron los dos pacientes seleccionados previamente y los resultados obtenidos fueron los siguientes:

Paciente	Predicción Final	Resultados Clasificación modelos	
		1 = Falla	0 = Cura
1	1 = Falla	5	0
2	0 = Cura	2	3

Tabla 13: ***Resultados evaluación integración de modelos.***
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Observamos que, en el caso del paciente uno, los cinco modelos lo clasificaron como 1, que corresponde a "falla" (no cura), y esta clasificación fue correcta. Para el paciente dos, tres de los cinco modelos lo etiquetaron como "cura", mientras que dos lo clasificaron como "falla". Al analizar en detalle, notamos que las redes neuronales convolucionales fueron las que clasificaron al segundo paciente como "falla", aunque debía ser clasificado como "cura". A pesar de esto, dado que tres de los cinco modelos lo clasificaron correctamente como "cura", el paciente finalmente fue correctamente asignado a la clase apropiada.

7. DISCUSIÓN DE RESULTADOS

En el transcurso de la investigación, se logró con éxito la consecución de todos los objetivos propuestos. El objetivo general de esta investigación consistía en predecir el desenlace terapéutico de la leishmaniasis mediante el uso de modelos de aprendizaje automático basados en fotografías de lesiones e información del transcriptoma. Esté fue abordado ejecutando cada uno de los procesos relacionados a los objetivos específicos. En primera instancia, se revisaron detalladamente los conjuntos de datos tanto de transcriptoma como de imágenes de lesiones. Con estos seleccionamos las mejores características para la construcción de los modelos SVM, K-Vecinos, Árboles de decisión, Bosque Aleatorio y las redes neuronales convolucionales con arquitecturas VGG16 y VGG19.

Para nuestro primer objetivo específico relacionado a la preparación de las fotografías de lesiones y la información transcriptómica de los pacientes, se presentaron unos retos importantes debido a la volumetría que tenían estos sets de datos. Por un lado, teníamos los transcriptomas con una dimensionalidad de alrededor 17000 genes, pero únicamente de 32 pacientes y por el otro lado teníamos las imágenes de lesiones de tan solo 30 pacientes donde las divisiones de los conjuntos de datos de entrenamiento y prueba requirieron el uso de múltiples técnicas, tanto de reducción de la dimensionalidad para los transcriptomas como incremento de las imágenes para abordar las redes convolucionales. Estas técnicas fueron utilizadas de forma satisfactoria e impactaron positivamente los resultados.

Como tercer objetivo específico buscamos modelar los datos relacionados con el desenlace terapéutico de los pacientes para lograrlo cada uno se construyó teniendo el objetivo estatus relacionado a cada uno de los pacientes donde cada uno estaba categorizado como cura o falla (no cura) y por lo tanto cada modelo tuvo un enfoque de categoría binaria representando 1 como falla y 0 como cura.

El cuarto objetivo estaba asociado a la optimización de los modelos de aprendizaje automático, donde para uno de los construidos, se partió de un modelo base con el fin de comparar las optimizaciones realizadas al obtener los mejores hiperparámetros con los que se tuvieron los mejores resultados. Este objetivo estaba directamente enlazado al siguiente donde evaluamos para cada uno el ajuste y la capacidad predictiva de los modelos utilizando la integración por votación que ofrece diversos beneficios, entre los cuales se destacan: mejora de la precisión, adaptabilidad y capacidad de generalización, optimización de recursos, entre otros.

La integración de los modelos fortaleció significativamente los resultados, superando las

limitaciones inherentes que se puedan llegar a presentar en la evaluación individual de los modelos de información transcriptómica y las fotografías de lesiones. Esta integración ha proporcionado unos resultados más robustos, consistentes y con mayor precisión.

La ejecución de cada proceso desarrollado en la presente investigación proporciona información valiosa sobre los posibles resultados que tendrá un paciente con diagnóstico de leishmaniasis que está siendo tratado, ya que permite crear un camino de posibilidades para identificar el desenlace terapéutico lo que puede convertirse en una potencial herramienta para aquellos que combaten esta enfermedad a diario.

8. CONCLUSIONES Y TRABAJOS FUTUROS

8.1 CONCLUSIONES

Los resultados obtenidos sugieren que tanto los modelos de aprendizaje automático como las redes neuronales exhiben una notable capacidad para abordar problemas de clasificación binaria. Esta capacidad se revela especialmente beneficiosa en el contexto de nuestra investigación, donde se busca predecir el desenlace terapéutico (cura o falla) a partir de fotografías de lesiones e información del transcriptoma.

Tanto los modelos base como aquellos optimizados mediante la estimación de hiperparámetros han demostrado eficacia al proporcionar estimaciones precisas, según las diversas métricas reportadas. La coherencia en las predicciones de los diferentes modelos utilizados permite extrapolar los resultados, convirtiendo la investigación en una herramienta significativa para respaldar decisiones clínicas en el CIDEIM.

Cabe destacar la interacción con diversas técnicas y modelos a lo largo del desarrollo de la investigación, permitiendo la ejecución de procesos de limpieza, reducción de dimensionalidad e incremento de datos incluso con conjuntos pequeños. La integración de modelos se erige como un paso crucial en la conclusión de esta investigación, seleccionando modelos en función de su rendimiento general y su comportamiento específico con genes seleccionados mediante técnicas de reducción de dimensionalidad. Este enfoque resultó en cinco modelos para la integración, facilitando la aplicación del método de ensamble por votación.

Es fundamental subrayar que la metodología adoptada es flexible y puede adaptarse y mejorar según los recursos disponibles, la disponibilidad de datos y las particularidades del estudio. Además, se enfatiza la importancia de cumplir con los requisitos éticos y regulaciones correspondientes a lo largo de todo el proceso de investigación, tales como transparencia y honestidad en el manejo de la información, consentimiento informado de las personas que hicieron parte de la investigación, consistencia y coherencia evitando sesgos, manipulación de datos o cualquier otra practica que pueda comprometer la integridad del estudio.

8.2 TRABAJOS FUTUROS

Dada la limitada cantidad de fotografías de lesiones e información del transcriptoma utilizados en el desarrollo de esta investigación, sería altamente beneficioso llevar a cabo evaluaciones de los modelos con nuevos casos que puedan surgir. En caso necesario, considerar la posibilidad de reentrenar los modelos se vuelve esencial, con el propósito de obtener predicciones más robustas y consistentes.

En cuanto a la integración del modelo, sería provechoso explorar otras técnicas de ensamble y comparar los resultados con la propuesta actual de la investigación. Además, se sugiere evaluar su rendimiento utilizando conjuntos de datos de prueba más amplios, lo que permitiría extrapolar las métricas obtenidas y proporcionar una visión más completa de su funcionamiento.

Dado que los modelos de aprendizaje automático e inteligencia artificial experimentan constantes actualizaciones y mejoras, resulta crucial implementar técnicas actualizadas sobre los datos trabajados. Este enfoque busca potenciar la capacidad predictiva de los modelos y mantenerlos alineados con los avances en la disciplina.

En resumen, este trabajo de grado no solo presenta hallazgos significativos, sino que también abre diversas vías de investigación, ofreciendo oportunidades para profundizar y mejorar los modelos estimados en futuras investigaciones.

9. ANEXOS

Durante el proceso de investigación y desarrollo de las redes neuronales, se han construido múltiples gráficos de las funciones de pérdida correspondientes a cada una de las arquitecturas implementadas. Estos gráficos permitieron comprender el desempeño y la convergencia de las redes en diferentes etapas del entrenamiento.

Es importante tener en cuenta que, aunque realizamos un exhaustivo análisis inicial, la ejecución actual puede presentar ciertas variaciones respecto a los resultados previamente obtenidos. Estas variaciones pueden surgir debido a diversos factores y por lo tanto es crucial mencionar que la escasez del conjunto de validación puede llevar a cambios significativos en el comportamiento de las redes.

Por lo tanto, al interpretar los resultados de las funciones de pérdida, debemos ser conscientes de estas posibles variaciones y considerarlas dentro del contexto de nuestra investigación. La adaptabilidad y la capacidad para ajustar los enfoques en función de nuevas observaciones y análisis son fundamentales para avanzar en la mejora continua de las redes neuronales propuestas en la presente investigación.

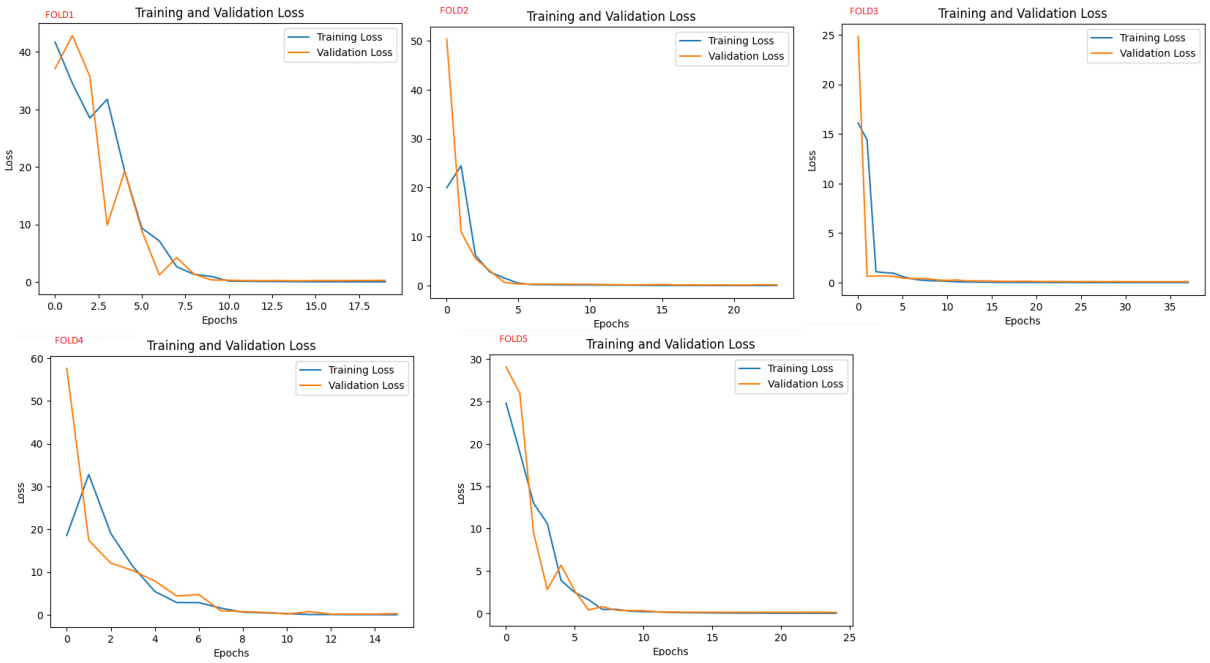


Ilustración 9: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***
Entrenamiento del modelo base por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

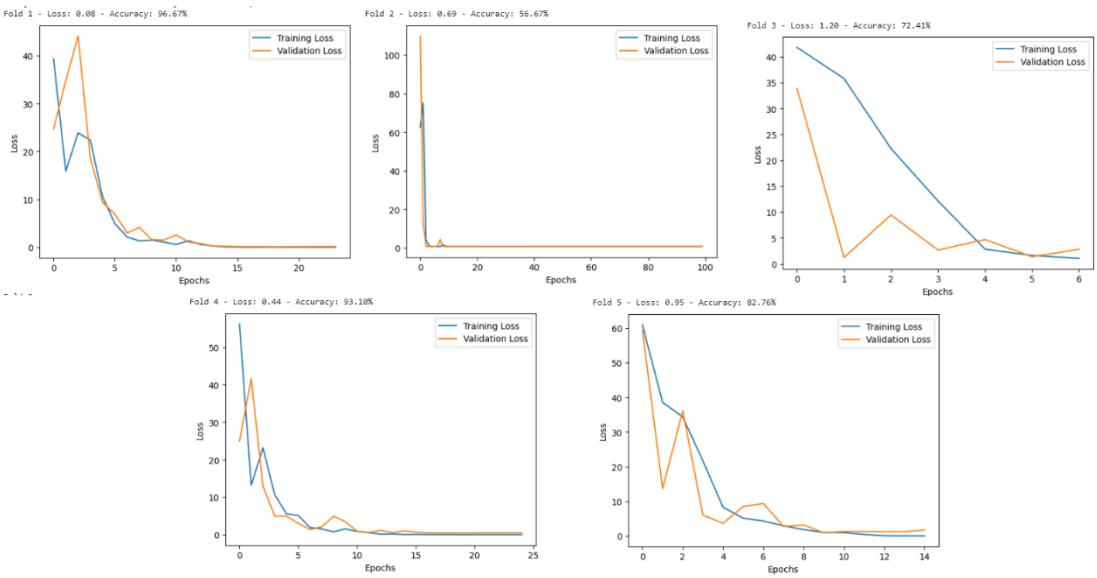


Ilustración 10: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***

Entrenamiento del modelo base optimizado por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

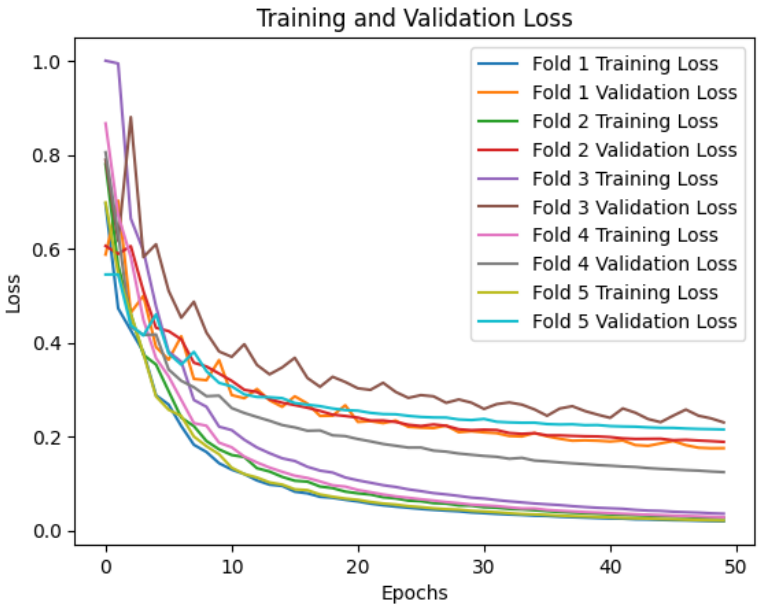


Ilustración 11: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***
Entrenamiento de la red vgg16 por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

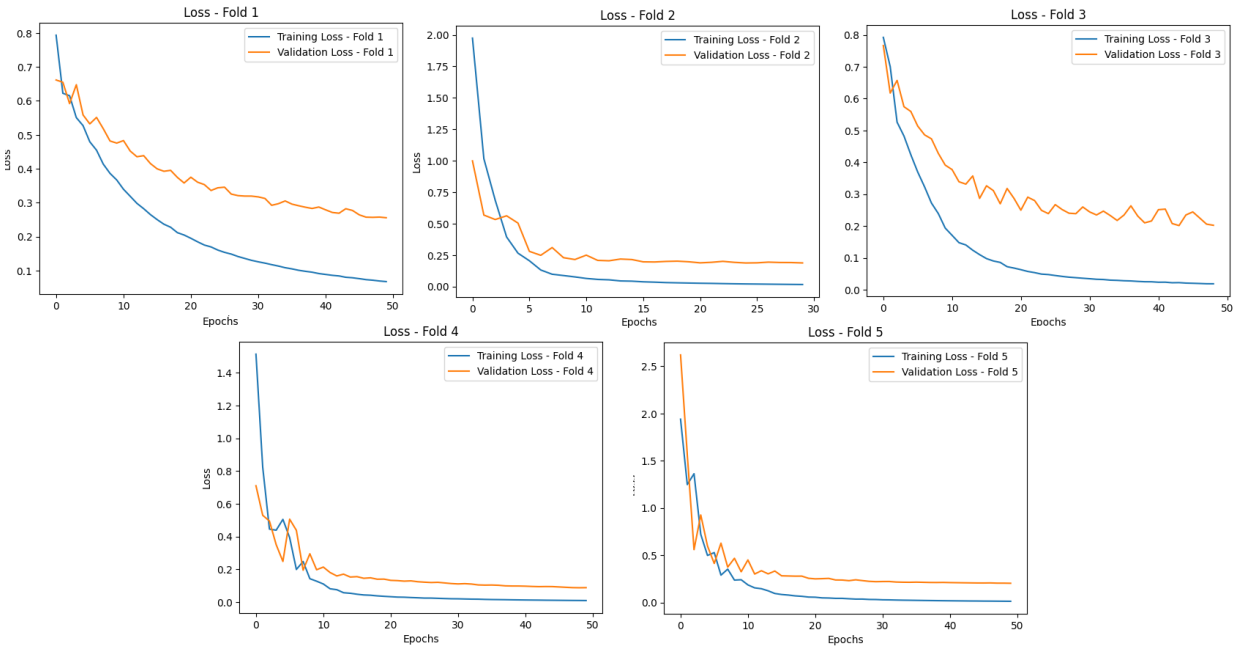


Ilustración 12: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***
Entrenamiento de la red vgg16 Optimizada por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

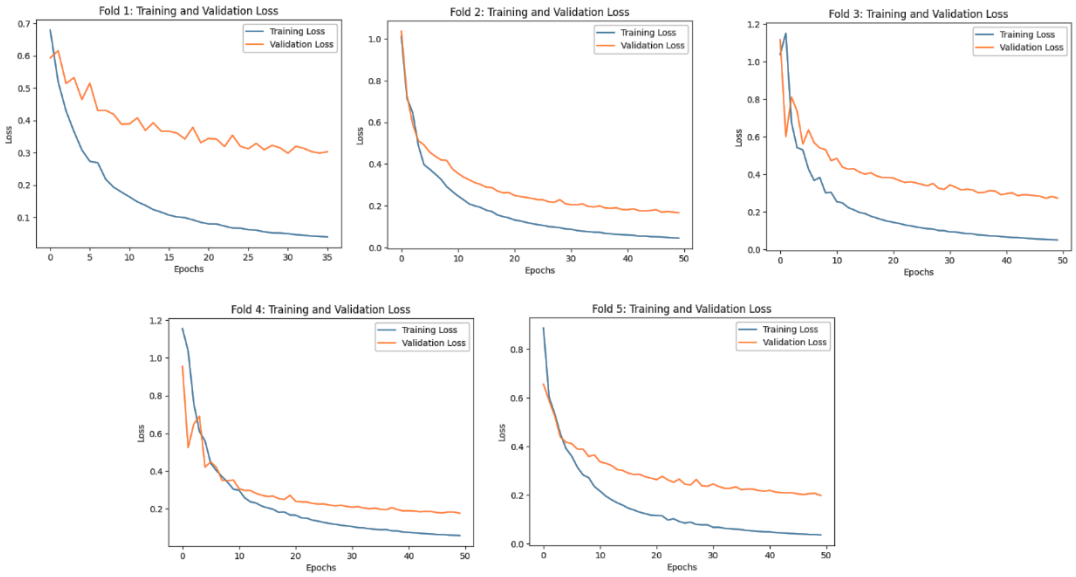


Ilustración 13: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***
Entrenamiento de la red vgg19 por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

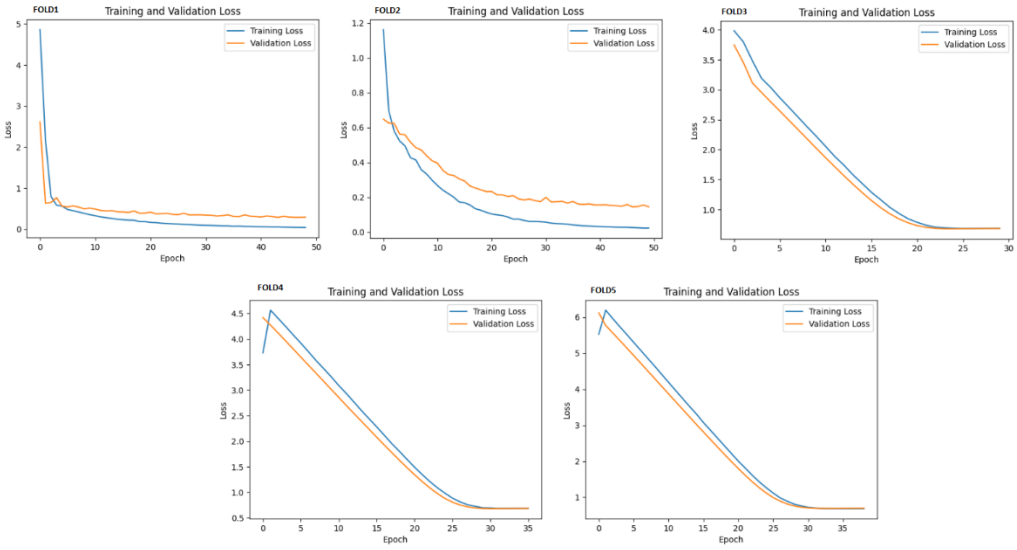


Ilustración 14: ***Función de pérdida para conjunto de datos de entrenamiento y validación por cada época.***
Entrenamiento de la red vgg19 Optimizada por cada fold.
Fuente: Elaboración propia a partir de estimaciones con datos del CIDEIM.

Repositório: https://github.com/MarioJArrieta/Leismaniasis
GitHub CLI: gh repo clone MarioJArrieta/Leismaniasis

10. REFERENCIAS BIBLIOGRÁFICAS

[1] P. Desjeux, «Leishmaniasis: current situation and new perspectives,» Comparative immunology, microbiology and infectious diseases, 2014, pp. vol. 27, no. 5, pp. 305-318.

[2] F. Norouzinezhad, F. Ghaffari, A. Norouzinejad, F. Kaveh y M. M. Gouya, «Cutaneous leishmaniasis in Iran: results from an epidemiological study in urban and rural provinces,» Asian Pacific journal of tropical biomedicine, 2023, pp. vol. 6, no. 7, pp. 614-619.

[3] J. Alvar, I. D. Vélez, C. Bern, M. Herrero, P. Desjeux y J. Cano, «Leishmaniasis worldwide and global estimates of its incidence,» PloS one, vol. 7, no. 5, 2012.

[4] L. P. Blanco, «Genómica y transcriptómica comparativa en cepas de Leishmania de Colombia,» 2019. [En línea]. Available: https://repository.urosario.edu.co/bitstream/handle/10336/20397/PatinoBlanco-LuzHelena-2019.pdf.pdf?sequence=1&isAllowed=y.

[5] J. Fraga, N. Veland, A. M. Montalvo, N. Praet, A. K. Boggild, B. M. Valencia, J. Arevalo, A. Llanos, J. C. Dujardin y G. Van der Auwera, «Accurate and rapid species typing from cutaneous and mucocutaneous leishmaniasis lesions of the New World,» de *Diagnostic microbiology and infectious disease*, 2012, pp. vol. 74, no. 2, pp. 142–150.

[6] Salud Global, «División de Enfermedades Parasitarias y Malaria,» 14 Febrero 2020. [En línea]. Available: http://www.cdc.gov/parasites/leishmaniasis/.

[7] O. P. d. Salud, « Leishmaniasis,» [En línea]. Available: https://www.paho.org/es/temas/leishmaniasis.

[8] Ministerio de salud y protección social, «Leishmaniasis,» [En línea]. Available: https://www.minsalud.gov.co/salud/publica/PET/Paginas/Leishmaniasis.aspx.

[9] B. L. Herwaldt, «Leishmaniasis,» *Lancet,* pp. vol. 354, no. 9185, pp. 1191–1199, 1999.

[10] F. Real, R.O. Vidal, M. F. Carazzolle, J. M. Mondego, G. G. Costa, R. H. Herai, M. Wurtele, L. M. Carvalho, R. Carmona y R. A. Mortara, «he genome sequence of Leishmania (Leishmania) amazonensis: functional annotation and extended analysis of gene models,» *DNA research,* Vols. %1 de %2vol. 20, no. 6, p. pp. 567–581, 2012.

[11] N. Marquis, B. Gourbal, B. P. Rosen, R. Mukhopadhyay y M. Ouellette, «Modulation in aquaglyceroporin AQP1 gene transcript levels in drug-resistant Leishmania,» *Molecular microbiology,* Vols. %1 de %2vol. 57, no. 6, pp. pp. 1690-1699, 2005.

[12] E. Torres Guerrero, M. R. Quintanilla , J. Ruiz y R. Arenas, «Leishmaniasis: a review,» *F1000Research,* Vols. %1 de %2vol. 6, , p. pp. 750, 2017.

[13] F. Vargas, E. Torres y M. R. Quintanilla, «Leishmaniasis en México,» *Med Cutan Iber Lat Am,* Vols. %1 de %2vol. 39, no. 4, , pp. pp. 163-183, , 2011.

[14] S. Muvdi y C. Ovalle , «Leishmaniasis mucosa: una enfermedad olvidada, descripción e identificación de especies en 50 casos colombianos,» *biomedica,* Vols. %1 de %2vol. 39, no. 2, pp. pp. 58-65, 2019.

[15] Y. Hashiguchi, E. L. Gomez, H. Kato, L. R. Martini, L. N. Velez y H. Uezato, «Diffuse and disseminated cutaneous leishmaniasis: clinical cases experienced in Ecuador and a brief review,» *Tropical medicine and health,* Vols. %1 de %2vol. 44, no. 2, 2016.

[16] M. L. Martínez, «Estructura y función del ADN y de los genes. I Tipos de alteraciones de la función del gen por mutaciones,» Vols. %1 de %2Volume 36,, nº 5, pp. 273-277, 2010.

[17] Instituto Nacional de Investigación del Genoma Humano, «genome,» 27 Septiembre 2019. [En línea]. Available: https://www.genome.gov/es/about-genomics/fact-sheets/Transcriptoma.

[18] I. Abadías, A. Diago, P. A. Cerro, A. M. Palma y Y. Gilaberte, «Leishmaniasis cutánea y mucocutánea,» *Actas Dermo-Sifiliográficas,* Vols. %1 de %2vol. 112, no. 7, pp. pp. 601-618, 2021.

[19] R. Gurrola y J. G. Rodriguez, Ciencia de los Datos, Propuestas y casos de uso, Universidad Pedagógica de Durango, 2020.

[20] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers,» *IBM,* vol. 3, nº 3, pp. 210-229, 1959.

[21] Y. Saeys, I. Inza y P. Larrañaga, «Una revisión de técnicas de selección de características en bioinformática,» *Bioinformática,* vol. 23, p. 2507–2517, 2007.

[22] I. Guyón, J. Weston, S. Barnhill y V. Vapnik, «Selección de genes para la clasificación del cáncer utilizando máquinas de vectores de soporte,» vol. 46, p. 389–422, 2002.

[23] M. Bamorovat, I. Sharifi, E. Rashedi, A. Shafiian, F. Sharifi, A. Khosravi y A. Tahmouresi, «A novel diagnostic and prognostic approach for unresponsive patients with anthroponotic cutaneous leishmaniasis using artificial neural networks,» *PLoS One,* vol. 16, nº 5, 2021.

[24] N. Shabanpour, S. V. Razavi, A. Sadeghi , C. Soo y T. Abuhmed, «ntegration of machine learning algorithms and GIS-based approaches to cutaneous leishmaniasis prevalence risk mapping,» *International Journal of Applied Earth Observation and Geoinformation,,* vol. 112, nº 3, 2012.

[25] H.A Pabón, I. Torres, M.Y. Parada y C.M. Durán, «Desarrollo de un sistema para la detección de leishmaniasis cutánea utilizando inteligencia artificial,» Universidad de Pamplona, Colombia, 2023.

[26] S. D. Jorge, «ANÁLISIS DE VARIANZA,» *Rev Chil Anest,* vol. 43, pp. 306-310, 2014.

[27] I. Jolliffe, Principal Component Analysis, Springer, 2002.

[28] R. A. Johnson y D. W. Wichern, Applied Multivariate Statistical Analysis, Pearson, 2007.

[29] J. Shlens, A Tutorial on Principal Component Analysis, arXiv preprint arXiv:1404.1100, 2014.

[30] J. E. Jackson, A User's Guide to Principal Components, John Wiley & Sons, 1991.

[31] D. Carl, Matrix Analysis and Applied Linear Algebra, 2020: Society for Industrial and Applied Mathematics (SIAM).

[32] G. A. Betancourt, «Las Máquinas de soporte Vectorial (SVMs),» *Scientia et Technica Año XI,* vol. 27, p. 67, 2005.

[33] J. Burges, «A Tutorial on Support Vector Machines for Pattern Recognition,» Boston, Bell Laboratories, Lucent Technologies, Kluwer Academic, 1996, pp. 1-43.

[34] T. Hastie, R. Tibshirani y J. Friedman, he Elements of Statistical Learning, Springer, 2009.

[35] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2014.

[36] T. Cover y P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory, 1967.

[37] J. Han, M. Kamber y J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.

[38] L. Breiman, «Random Forests,» de *Machine learning,*, 2001, pp. 5-32.

[39] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» Nature, 2015, pp. 436-444.

[40] Redacción KeepCoding, «keepcoding,» 22 Mayo 2023. [En línea]. Available: https://keepcoding.io/blog/arquitectura-vgg16-vgg19-deep-learning/#Arquitectura_VGG16_y_VGG19_en_Deep_Learning.

[41] J. Cadenas y C. Garrido, Soft Computing en ensambles basados en boosting, badding y randomforest, Dpto. Ingenieria de la información y las comunicaciones.

[42] M. Simic, «baeldung,» 14 Febrero 2023. [En línea]. Available: https://www.baeldung.com/cs/hard-vs-soft-voting-classifiers..

[43] D. M. Kelmansky, Análisis Exploratorio y Confirmatorio de Datos de Experimentos de Microarrays, Dpto. de Matemática - Instituto de Cálculo, 2003.

[44] V. D. l. Rosa, «EL PROCESO DEL TRATAMIENTO MÉDICO, Compilación: Dr. E. Víctor De la Rosa Morales Neurólogo Pediatra Maestro en Ciencias de la Salud Pública,» *Investigación y desarrollo en salud,* vol. 4, nº 5, pp. 28-29, 2015.

[45] T. M. Mitchell, «Does machine learning really work?,» *AI Magazine,* vol. 18, nº 3, p. 11, 1997.

[46] W. Lian, G. Nie, B. Jia, D. Shi, Q. Fan y Y. Liang, «An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning,» *Hindawi Mathematical Problems in Engineering,* p. 15, 2020.

[47] M. Tuiran, «Inteligencia Artificial en Relación con la Medicina,» *Revista Ingenierías USBMed,* vol. 12, nº 2.

[48] A. M. Rubioa, M. O. Mohameda, D. R. Ferreira, A. Arroyo, P. Mesa y J. J. Hernández , «La leishmaniasis visceral,» *Medicina integral,* vol. 36, nº 8, pp. 294-299, 2000.

[49] A. K. Cruz y F. Freitas, «Genome and transcriptome analyses of Leishmania spp.: opening Pandora's box,» *Current*

Opinion in Microbiology, vol. 52, pp. 64-69, 2019.

[50] G. Rodriguez, F. Jaramillo y J. G. Chalela, «LA BIOPSIA DE PIEL,» *BlOMEDlCA,* vol. 7, nº 1, 1987.

[51] P. C. Anderson, «Skin biopsy,» *JAMA,* vol. 201, nº 1, 1967.

[52] A. Géron, Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow, ANAYA MULTIMEDIA, 2020.

[53] M. Bamorovat, I. SharifiID, E. Rashedi, A. Shafiian, F. Sharifi, A. Khosravi y A. Tahmouresi, «A novel diagnostic and prognostic approach for unresponsive patients with anthroponotic cutaneous leishmaniasis using artificial neural Networks,» *PLoS One,* vol. 16, nº 5, 2021.

[54] Ministerio de Salud y Protección Socia, «minciencias,» Abril 2018. [En línea]. Available: https://minciencias.gov.co/content/centro-internacional-entrenamiento-e-investigaciones-medicas-cideim.

[55] ATSDR, «atsdr,» Octubre 2019. [En línea]. Available: https://www.atsdr.cdc.gov/es/toxfaqs/es_tfacts23.html#:~:text=El%20antimonio%20puede%20tener%20efectos,musculares%20y%20en%20las%20articulaciones.

[56] A. M. .. Craig y A. Rachel, «Advanced and Multivariate Statistical Methods: Practical Application and Interpretation,» Routledge, 2021, p. 15.

[57] G. Pablo, Regresion lineal y Análisis de Varianza (ANOVA), Universitat Jaume I de Castellón, 2015/2016.

[58] Neurohive, «neurohive.io,» 20 Noviembre 2018. [En línea]. Available: https://neurohive.io/en/popular-networks/vgg16/.

[59] V. Galán y E. Castro, «APLICACIÓN DE LA METODOLOGÍA CRISP-DM A UN PROYECTO DE MINERÍA DE DATOS EN EL ENTORNO UNIVERSITARIO,» Madrid, Universidad Carolos III, 2015.

[60] Organización Panamericana de la Salud, «Leishmaniasis,» [En línea]. Available: https://www.paho.org/es/temas/leishmaniasis. [Último acceso: 15 Julio 2023].

[61] M. d. M. C. Noriega, «Factores asociados a falla terapéutica en niños y adultos con Leishmaniasis cutánea en tres zonas endémicas de Colombia,» Universidad del Valle, Santiago de Cali, 2015.

[62] V. I. L. L, «Thermotherapy effective and safer than miltefosine in the treatment of cutaneous leishmaniasis in Colombia,» Rev Inst Med Trop, Sao Paulo, 2013.

[63] M. P. R, «Treatment failure in children in a randomized clinical trial with 10 and 20 days of meglumine antimonate for cutaneous leishmaniasis due to Leishmania viannia species,» Am J Trop Med Hyg, 2001.

[64] BioDatev, «LOG FOLD CHANGE: MIDIENDO LAS DIFERENCIAS EN LA EXPRESIÓN GÉNICA,» [En línea]. Available: https://biodatev.com/log-fold-change/. [Último acceso: 15 Diciembre 2023].