Detección de presencia y ausencia de Parkinson en audios de voz

JUAN DAVID SANDOVAL RAMIREZ

ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

FACULTAD DE INGENIERÍA

UNIVERSIDAD DEL VALLE

SANTIAGO DE CALI

2025

Detección de presencia y ausencia de Parkinson en audios de voz

JUAN DAVID SANDOVAL RAMIREZ

**TRABAJO PRESENTADO PARA OPTAR
POR EL TÍTULO DE INGENIERO
ELECTRÓNICO**

**DIRECTOR
HUMBERTO LOAIZA CORREA, Ph.D.**

Grupo de Investigación Percepción y Sistemas Inteligentes

ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

FACULTAD DE INGENIERÍA

UNIVERSIDAD DEL VALLE

SANTIAGO DE CALI

2025

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

Nota de Aceptación

Director

Codirector

Jurado

Jurado

AGRADECIMIENTOS

Inicio estos agradecimientos hacia mi mamá Amparo, mi papá Ricardo, mi hermano Ricardo y mi pareja actual, quienes fueron pieza fundamental en este proceso, confiaron en mí desde el primer momento que ingresé a la universidad hasta el día de la presentación de este proyecto, gracias a ellos he logrado culminar esta etapa.

Gracias al profesorado que me acompañó durante mi proceso académico, haciendo mención especial a mi director Humberto Loaiza, que estuvo brindándome su ayuda para el cumplimiento de este proyecto.

A mis compañeros que estuvieron desde el primer minuto Carlos Moreno y Pablo Moreno, que me apoyaron desde un comienzo en mi proceso académico, así como en mi vida personal. También a aquellos que se unieron en el camino como Christian Garcia, Julian Sanchez y Juan David Cardenas, quienes también hicieron de la universidad un lugar para disfrutar entre amigos.

Además, agradecer a mi familia, quienes también fueron parte de este lindo proceso, así como a mis amigos externos a la academia que estuvieron brindándome su apoyo emocional en momentos complicados.

Por último, agradecer a la universidad por permitirme vivir esta experiencia y poder culminar mis estudios en ella.

Resumen

El objetivo de este trabajo de grado es la clasificación de presencia y ausencia de Parkinson mediante grabaciones de audio, que fueron extraídas de la base de datos IPVVS con grabaciones de voz en italiano. Además, se realiza la investigación de antecedentes nacionales e internacionales que hayan implementado procesos similares a los previstos en este proyecto, así como la revisión de un marco teórico que provea una contextualización de la aplicación en cuestión.

El sistema propuesto consta, primeramente, de la etapa de preprocesamiento a los audios de la base de datos, que incluyó eliminación de silencios, remuestreo de frecuencia, filtrado y aumento de datos, mediante ventanas. En segundo lugar, la etapa de extracción de características y de espectrogramas, basados en los antecedentes consultados, además se aplicaron técnicas como Wrappers y Feature Selection para la selección de las características más discriminantes y la posterior normalización de los datos. Luego, con esta información se implementaron las máquinas de aprendizaje superficial como Random Forest (RF), Decision Trees (DT), Suppor Vector Machines (SVM) y K-Nearest Neighboors (KNN), teniendo dos maneras de clasificación. Finalmente, se implementaron dos redes neuronales convolucionales (CNN), como ResNet 18 y EfficientNet-B0, que fueron previamente entrenadas con ImageNet y aplicadas en este proyecto por medio de TransferLearning, debido al tamaño de la base de datos.

En cuanto a los resultados obtenidos a través del sistema descrito, las máquinas de aprendizaje superficial lograron un promedio de exactitud en clasificación de 93.60%, destacándose KNN como el mejor clasificador con 96.38%, mientras que para el caso de las máquinas CNN el mejor desempeño registrado fue 98.56% por parte de EfficientNet-B0. Finalmente, se realiza la discusión de los resultados obtenidos y se establecen conclusiones con base en estos.

Palabras Clave: Aprendizaje superficial, Parkinson, Procesamiento de señales de voz, Reconocimiento de patrones, Red neuronal.

Abstract

The objective of this thesis is to classify presence and absence of Parkinson's disease on audio recordings, which were extracted from the IPVVS database that contains voice recordings in Italian. Additionally, the research of national and international studies related focusing on similar process to those proposed in this project, as well as the review of a theoretical framework that provides a contextualization for the application in question.

The proposed system consists begins with a preprocessing stage to the audios from the database, which included elimination of silences, frequency resampling, filtering and data augmentation using windows. Next, the stage of feature extraction and spectrograms, based on insights from the literature, also Wrappers and Feature Selection were applied to select the most discriminative features and data normalization. Then, with this information, shallow learning machines were implemented including Random Forest (RF), Decision Trees (DT), Suppor Vector Machines (SVM) and K-Nearest Neighboors (KNN), using two ways to classify. Finally, two convolutional neural networks (CNN) such as ResNet 18 and EfficientNet-B0 were implemented previously trained by ImageNet database and applied in this project by TransferLearning, due to the size of the database.

Regarding the results obtained from the described system, the shallow learning models achieved an average classification accuracy of 93.60%, with KNN standing out as the best classifier with 96.38%, meanwhile CNN machines cases, the best performance recorded was 98.56% by EfficientNet-B0. Finally, the results were discussed and conclusions are drawn based on the findings.

Key Words: Artificial neural network, Parkinson’s disease, Pattern recognition, Shallow learning, Voice signal processing

1. Introducción

La enfermedad de Parkinson (PD), es un trastorno neurodegenerativo que afecta progresivamente el control motor en el cerebro, y presenta diferentes sintomatologías a raíz de este[1], por estos motivos, la detección de esta enfermedad por medio de diferentes alternativas resulta fundamental para la vida de las personas que la padecen. En este contexto, la detección mediante el análisis de señales de voz puede llegar a ser una herramienta prometedora para el diagnóstico de esta enfermedad, debido a que las personas que la padecen presentan alteraciones en la voz las cuales pueden ser detectadas[2].

Este trabajo de grado se centra en la detección de la enfermedad de Parkinson a partir de grabaciones de audio, dispuestas en una base de datos de acceso público. Para las máquinas de aprendizaje superficial se tuvo que aplicar un preprocesamiento de señales de audio, que incluyó diferentes procesos como filtrado, normalización, etc. Además de la extracción de características discriminantes para esta aplicación de diferentes tipos como centrales, espectrales, entre otras. Los modelos de aprendizaje superficial implementados fueron K-Nearest Neighbors (KNN) con 96.38% de exactitud, Suport Vector Machine (SVM) 95.65% de exactitud, Decision Tree (DT) con 95.65% y Random Forest (RF) con 94.93% de exactitud, con el fin de dar una visión más amplia acerca de este tipo de modelos.

Por otro lado, para el aprendizaje profundo se tuvo que extraer una representación bi-dimensional del audio mediante espectrogramas a los audios previamente preprocesados, los cuales fueron empleados como entrada de las máquinas implementadas. Los modelos de aprendizaje profundo se realizaron con *Transfer Learning* mediante dos arquitecturas de redes neuronales convolucionales como ResNet18 con 95.49% de exactitud Y EffcientNet-B0 con 98.56% de exactitud. La combinación de estos enfoques permite evaluar la eficacia de los diferentes métodos de clasificación implementados en la detección de PD.

1.1 Descripción del trabajo

Para el desarrollo de este proyecto, primero se realizó una contextualización de la problemática a abordar mediante el marco teórico, así como la revisión y análisis de antecedentes altamente relacionados o similares. Además, la base de datos seleccionada se recortó de tal forma que se adaptara a esta aplicación, siguiendo con los procesos de preprocesamiento similares a los encontrados en la literatura analizada, realizando ciertas variaciones que se acoplaran a lo propuesto en este trabajo con el fin de obtener grabaciones de audio más discriminantes para el proceso de clasificación de Parkinson (PD).

En la etapa de preprocesamiento, se aplicó un filtro paso bajo de acuerdo con el análisis de frecuencias de los audios. Posteriormente, se recortaron los audios para estandarizar su duración y se eliminaron los silencios presentes a partir de un umbral de silencio específico. Con el fin de aumentar los datos se dividieron los audios mediante segmentación por ventanas. Por otro lado, para las máquinas de aprendizaje profundo, se tuvieron que extraer los espectrogramas de los audios previamente preprocesados, para emplearlos como datos de entrada. Lo anterior y las etapas posteriores, fueron realizadas en Python con la ayuda de las librerias de Numpy, Scipy, TensorFlow, SckitLearn, entre otras.

Para la etapa de extracción de características, se calcularon diferentes tipos de variables como ceptrales, espectrales, entre otras. Esto con el fin de discriminar los datos para su posterior proceso de clasificación, que fue hecho con las máquinas de aprendizaje superficial de SVM, KNN, RF y DT. Adicionalmente se implementaron las máquinas de aprendizaje profundo que fueron dos CNN con dos arquitecturas prestablecidas y preentrenadas por medio de *Transfer Learning*.

Finalmente, se analizaron los resultados obtenidos en cada una de las etapas anteriores, lo que permitió concluir acerca del desempeño de los modelos y rendimiento en clasificación de pacientes con PD.

1.2 Planteamiento del Problema y Objetivos

1.2.1 Planteamiento del problema

El Parkinson (PD, Parkinson´s Desease) es una afección cerebral neurodegenerativa que a día de hoy no existe cura, aunque algunos tratamientos pueden reducir notablemente los síntomas [1]. En los últimos 25 años, el número de pacientes con PD se ha duplicado, superando los 10 millones de diagnósticos y siendo la causa de más de un millón de fallecimientos en su historia[3].

La detección de esta enfermedad puede ser un reto desafiante, puesto que los síntomas de los pacientes suelen confundirse con envejecimiento normal y el diagnostico, en su mayoría, se detecta en las etapas más avanzadas, lo ideal sería la detección en la etapa prodrómica [4][5]. Actualmente, existen ciertos exámenes para la detección como resonancias magnéticas, ecografías transcraneales y tomografías, que suelen ser bastantes costosas y que en cuanto a eficacia pueden presentar limitaciones. Por esta razón, ante la problemática se han desarrollado métodos alternativos principalmente basados en inteligencia artificial con el fin de reducir los costos de estos y aumentar la precisión de los diagnósticos [6][7].

Entre los métodos alternativos para la detección de presencia o ausencia de la PD se encuentra el análisis de los patrones de las señales de voz con máquinas de aprendizaje superficial (*Shallow learning*) y profundo (*Deep learning*). Ejemplos de desarrollos internacionales son descritos en [7][8] donde han alcanzado desempeños de exactitud y precisión mayores a 90%. En Colombia se encuentran algunos desarrollos, como los reportados en [9][10], que alcanzaron desempeños de 92% y 95%.

El proceso de detección de PD en señales de voz mediante algoritmos de aprendizaje superficial y profundo consta de cuatro etapas: La primera es la adquisición de señales, que requiere de la captura de audios de personas con y sin PD; La segunda es la etapa de preprocesamiento de audio, donde se eliminan ruidos y frecuencias no relevantes para la investigación; Como tercera etapa se encuentra la extracción y selección de características, donde se extraen estas y se identifican las más útiles para clasificación: Por último, la cuarta es la clasificación automática, donde se implementan modelos de aprendizaje para diferenciar las personas con y sin PD.

Considerando que el análisis de señales de audio para la detección de la presencia o ausencia de PD es un tema de interés mundial, abierto en investigación, donde no se han resuelto muchas de las dificultades que presentan las etapas de un sistema con este propósito, se formula la siguiente pregunta problematizadora: ¿Cómo implementar un sistema de reconocimiento de patrones con máquinas de aprendizaje automático para detectar la presencia o ausencia de la enfermedad de Parkinson en señales de audio?

En este proyecto, se explorarán y se implementarán procedimientos de análisis de señales de audio y reconocimiento de patrones, esto por medio de máquinas de aprendizaje. Para ello, se implementará una base de datos de acceso público, que lo provee una entidad educativa (como, por ejemplo, el reportado en [8]), con el fin de diseñar, implementar y finalmente evaluar diferentes algoritmos. Este proyecto, se enfocará en la clasificación de Parkison mediante señales de audios, desde una perspectiva netamente académica.

1.2.2 Objetivo General

Detectar la presencia o ausencia de la enfermedad de Parkinson en audios de voz mediante técnicas de procesamiento de señales digitales y reconocimiento de patrones con máquinas de aprendizaje.

1.2.3 Objetivos Específicos

- Seleccionar características de señales digitales de voz que favorezcan la diferenciación entre presencia y ausencia de la enfermedad de Parkinson.
- Implementar algoritmos de procesamiento digital de señales para la adecuación y extracción de información de los audios de voz que potencialicen la diferenciación entre presencia y ausencia de la enfermedad de Parkinson.
- Implementar dos máquinas de aprendizaje superficial para clasificar la presencia y la ausencia de la enfermedad de Parkinson en señales digitales de voz.
- Implementar dos máquinas de aprendizaje profundo para clasificar la presencia y la ausencia de la enfermedad de Parkinson en señales digitales de voz.

- Evaluar el desempeño de las máquinas de aprendizaje superficial y profundo en la detección de la presencia y ausencia de la enfermedad de Parkinson en señales digitales de voz.

2. Antecedentes y Marco Teórico

En este capítulo se presentan los antecedentes nacionales e internacionales con resultados y metodologías relevantes para el desarrollo de este trabajo. Asimismo, se presentan los conceptos necesarios para el entendimiento de las metodologías, técnicas y desarrollos implementados en cada parte del trabajo.

2.1 Antecedentes

El reconocimiento de presencia o ausencia PD mediante la voz han sido uno de los grandes objetos de investigación en los últimos años, ya que permiten una detección de la enfermedad sin realizar estudios muy específicos. Por esta razón, se investigaron algunos antecedentes relacionados con la temática a trabajar. Para esto se utilizó la base de datos IEEE y el repositorio de la Universidad del Valle. Cabe mencionar, que sólo se consultaron trabajos que fueron publicados entre los años 2019 y 2024, usando las palabras clave: Parkinson desease detection, Parkinson desease by voice. Antecedentes Nacionales

2.1.1 Antecedentes Nacionales

Este proyecto [9] no es un proyecto exclusivamente colombiano, sin embargo, Orozco Arroyave es uno de los autores, junto con otros autores internacionales de India y Alemania. El trabajo tenía como objetivo desarrollar y entrenar máquinas de aprendizaje superficial y profundo para la detección de PD entre diferentes idiomas, como el español y el italiano, además del género del paciente mediante múltiples experimentos. Para llevar a cabo estos procedimientos, se tuvieron en cuenta tres bases de datos, dos de ellas en español y una en italiano, cada una de ellas contaba con grabaciones de audios de las cinco vocales presentes en estos dos idiomas. En la etapa de extracción de características se implementaron dos métodos diferentes: Descomposición de modos variables (VMD) y el espectro de Hilbert (HS), extrayendo características como Jimmer, Shimmer, entre otras. Los clasificadores implementados fueron CNN, MLP, RF y SVM, como se menciona anteriormente estos clasificadores fueron entrenados con diferentes combinaciones de datos de prueba y entrenamiento, obteniendo un desempeño máximo de 80%, mientras que

cuando se trabajaban las bases de datos por separado o en el mismo idioma se alcanzaban exactitudes de hasta 95%. En términos generales el estudio obtuvo buenos resultados de manera independiente, mientras que, al combinar idiomas, se presentó una limitante, puesto que las métricas se vieron afectadas en cuanto a desempeño.

Siguiendo el mismo camino del trabajo anterior, en [11] este proyecto están involucrados autores extranjeros, como Orozco Arroyave, además comparte una de las bases de datos utilizados en el trabajo anterior. La base de datos incluye 100 pacientes, 50 de ellos con PD y los restantes HC. El objetivo de este proyecto era la clasificación de estos dos grupos de pacientes mediante el uso de espectrogramas de 40ms y 20ms. En la etapa de preprocesamiento, se implementó un filtro paso banda. Luego de esto se extrajo el espectrograma y se procedió a implementar un codificador CNN, seguido de una máquina de aprendizaje con capas de memoria a corto y a largo plazo (LSTM), implementando una arquitectura ResNet. Como resultado de haber aplicado esta máquina de aprendizaje profundo, se obtuvo una precisión en clasificación de 91.7%. Debido a la limitación de la base de datos, no se obtuvieron mejores resultados, y se dejó para posteriores proyectos. Sin embargo, se destaca facilidad de la adquisición de datos, puesto que los pacientes no necesitan de entrenamiento previo para la grabación de audios.

2.1.2 Antecedentes Internacionales

Este antecedente internacional [12] tuvo como objetivo clasificar pacientes con PD y sanos a través de técnicas de ML, un dato a resaltar es la cantidad de pacientes que pudieron realizar el proceso de grabaciones de audio, en total fueron 252 pacientes con 3 audios, cada uno entonando la vocal “A”, lo cual 188 de ellos con PD y los restante sanos. Para llevar a cabo este proyecto, se extrajeron 755 características, destacando 456 que se obtuvieron por medio de Tunable Q factor wavelet transform (TWQT). Debido a la distribución de pacientes, se realizó un balance de los datos mediante SMOTE. Los modelos de aprendizaje que se implementaron fueron SVM, KNN, DT, RF, Adaboost, Gradient Boosting y MLP. Para limitar el uso de características seleccionadas se empleó la máquina de soporte vectorial lineal (LinearSVC). Los resultados obtenidos de precisión en testeo, destacando Adaboost con 84.21% y MLP con 82.02%, por otro lado, en precisión se obtuvo con KNN un 93%, además la máquina con mejor resultado de F1 Score fue Adaboost con 89%. Por último, se realizó un entrenamiento empleando

combinaciones de dos máquinas con el fin de mejorar los resultados obtenidos donde se destacó Adaboost con LSVC con u 85.09% de precisión, mejorando el mejor resultado de que se obtuvo con los modelos independientes.

Este proyecto [7], tiene como objetivo la comparación de máquinas de aprendizaje superficial y profundo, específicamente KNN, RF, DT, Feed-Forward Network (FNN) and CNN. La base de datos utilizada en este trabajo es la UCI, que consiste en 195 grabaciones de audio de 31 pacientes. Se empleo técnicas de reducción de dimensionalidad de características como Recursive Feature Elimination (RFE) y SelectK-Best, además de RandomizedSearchCV para los hiperparámetros. Dentro de las características extraídas se encuentran 24 medidas biomédicas, destacando jitter, shimmer, entre otras. Para asegurar la dimensionalidad de la base de datos, se empleó SMOTE, y para el entrenamiento y testeo se implementó una división del 80%-20%. Para el proceso de clasificación se tomaron tres diferentes métodos:

1. Sin selección de características y sin RandomizedSearch: Los resultados destacados fueron FNN con una exactitud de clasificación de 88.19% y precisión de 98.12%, seguido por RF con 86.89% con precisión de 94.49%.

2. Con Randomized search y selección de características: FNN nuevamente destaco en exactitud con 92.12% y precisión de 96.98%, seguido por RF con 89.61% y 96.07% respectivamente.

3. Con RandomizedSearch únicamente: Se obtuvieron mejores resultados en FNN de 99.11% de exactitud y 99.96% de precisión, destacando sobre los otros modelos. Sin embargo, el costo computacional es una limitante, puesto que se emplean todas las características.

Este proyecto, publicado en 2023 [13], tiene como objetivo la detección de PD mediante el uso de dos bases de datos diferentes y diversos métodos para clasificación. En el proyecto se extrajeron 17 características, y se empleó un método de selección de características para obtener las mejores 8 mediante un clasificador Extra Tree, y un algoritmo genético. Para lograr este objetivo, se utilizaron máquinas de aprendizaje superficial como RF, KNN, XGBoost y regresión logisitca. En total, se implementaron 6 pruebas de clasificación con cada una de las máquinas, y 3 para cada base de datos: una con todas las características, otra con las seleccionadas por extra tree y una última con las seleccionadas mediante el

algoritmo genético. Los resultados obtenidos con la base de datos IPVS fueron bastante prometedores, puesto que se obtuvieron resultados mayores a 93% con el algoritmo genético, incluso hasta alcanzar 100%, aunque esto puede ser un dato de sobreajuste de la máquina. Por parte de la otra base de datos empleada, se obtuvieron solo resultados con un máximo de exactitud de 91.13% con regresión logística.

Este trabajo [5] tiene como principal objetivo la comparación entre modelos de aprendizaje superficial y profundo. Para esto, se utilizó una base de datos de cien personas, divididas a la mitad entre HC y pacientes con PD. En cuanto al preprocesamiento, se realizó un filtro paso bajo, normalización de los audios y se dividieron los audios en 100 ventanas de 12.5 ms. Además, cualquier audio que tuviera menos de 1.5 segundos fue cortado, resultando con 41 HC y 40 PD. Para las máquinas de aprendizaje superficial como LR y RF se extrajeron varios tipos de características espectrales como LPC, MFCC, CEP y LAR, obteniendo como resultado de área bajo la curva (AUC) máximo 73%. Mientras que para el trabajo con CNN mediante espectrogramas en escala de grises y colores se obtuvo un 97% de AUC, incrementando considerablemente esta métrica.

Este último proyecto [14] tiene como objetivo la comparación del aprendizaje superficial contra el aprendizaje profundo para la clasificación multiclase de PD, por medio de las máquinas SVM, KNN, Naive Bayes y CNN. Se creo una base de datos propia con 426 pacientes diferentes, divididos en avanzados activos, avanzados no activos, pacientes positivos y pacientes sanos. En cuanto al preprocesamiento de audio, se aplicó un filtro FIR. Para la extracción de características, se implementaron espectrogramas para la CNN, y para las máquinas de aprendizaje superficial se extrajeron 453 características. además, para la selección de características se emplearon el método wrappers, el algoritmo basado en la correlación (CFS) y mRMRS. Las máquinas de aprendizaje superficial se compararon de varias formas: biclase y multiclase. Para biclase, entre medio avanzado PD y PD temprano se obtuvo una exactitud del 85%, mientras que para multiclase incluyendo los pacientes sanos se obtuvo un máximo de 61%. Para CNN se obtuvieron resultados máximos de 82% en exactitud de clasificación en biclase, y para multiclase se obtuvo un 62% de exactitud.

2.1.3 Discusión

Se tuvieron en cuenta siete proyectos, con dos de ellos con autoría compartida colombiana y extranjera, los restantes completamente extranjeros, donde los procesos realizados en estos proyectos son similares y muy relacionados con este trabajo, aunque existen diferentes formas de abordar las máquinas de aprendizaje y varían de acuerdo con el proyecto. A pesar de la similitud, resulta relevante abordar las diferencias de los métodos empleados como lo son el preprocesamiento de los datos, el idioma de la base de datos, el tipo de audios, y las máquinas de aprendizaje empeladas para clasificación.

En cuanto al preprocesamiento de los audios, algunos de los antecedentes como [7] [12] no son muy específicos en cuanto a este apartado, o simplemente se tomó una base de datos dada en un archivo CSV con las características extraídas y se trabajaron a partir de aquí para las máquinas de aprendizaje. En estos dos proyectos, se emplearon el proceso de SMOTE para la preparación de los datos, sin embargo, existe una diferencia entre estas, pues, en [7] se tomaron únicamente 191 audios y 23 características, mientras que en [12] se tomaron 756 grabaciones y 755 características, siendo este el antecedente con la base de datos más extensa.

En cuanto a los demás proyectos revisados, el preprocesamiento de datos, en [9] se aplicaron un filtro espectral, un filtro pasa bajas de 12 kHz y un filtro FIR de 30 taps, por otro lado tenemos en [5] un filtro de 300 dB para hombres y 600 dB para mujeres. Como se puede observar los antecedentes no son muy específicos en cuanto a todos los procesos de preprocesamiento de audio, pero podemos observar un comportamiento similar en cuanto al realizar un filtrado paso bajo para la eliminación de ruido y frecuencias de no interés.

Con respecto al tema de la base de datos, el idioma es un aspecto relevante, puesto que realizar clasificación de PD entre idiomas es un proceso bastante complejo como se ve reflejado en [9] en cuanto a métricas de desempeño. En los proyectos empleados, se tiene en su mayoría bases de datos en italiano, inglés y español, sin embargo, uno de ellos es en turco. Teniendo en cuenta esto, cuando se trata de el mismo idioma y biclase podemos observar la mejora en el desempeño, puesto que el más bajo es de 84.21% en [12] obteniendo desempeños hasta 99.11% en [7]

Tabla 2-1 Tabla resumen de los procesos realizados y resultados obtenidos en los antecedes consultados

Referencia	Tamaño base de datos	Idioma	Preprocesamiento	Máquinas implementadas	Desempeño
[9]	a/e/i/o/u 100 pacientes (PC-GITA)	Español	No se especifica	CNN, MLP, RF, SVM	90% SVM
	a/e/i/o/u 40 pacientes (PC-GITA -IND)	Español	No se especifica	CNN, MLP, RF, SVM	95% CNN
	a/e/i/o/u 50 pacientes (IPVSS)	Italiano	No se especifica	CNN, MLP, RF, SVM	90% MLP
[11]	/a/ 100 pacientes (PC-GITA)	Español	Recorte audio 0.5 seg Filtro paso banda	CNN	91.7% LSTM
[12]	/a/ (3 veces) 252 pacientes	Turco	Balanceo de datos normalización de datos	SVM, LR, KNN, RF, DT, MLP, adaboost, Gradient boost	84.21 % adaboost
[7]	31 pacientes 195 grabaciones de voz	Ingles	Smote	KSVM, RF, DT, KNN, FNN	99.11 % FNN
[13]	/a/e/i/o/u/ 48 pacientes	Italiano	No se especifica	KNN, XGboost, RF, LR	98.58 % RF
	37 pacientes párrafo leído y dialogo espontaneo	Ingles	No se especifica	KNN, XGboost, RF, LR	98.09 % RF
[5]	/a/ 100 personas	Ingles	Eliminar silencio Filtro pasa bajo normalización Ventanas de tiempo	LR, RF, CNN	0.97 AUC

[14]	/e/ 266 pacientes	Italiano	Filtro paso bajo Normalization	KNN, SVM, CNN	85% KNN Biclase 62 % CNN Multiclase

2.2 Marco Teórico

El PD es un trastorno que es neurodegenerativo, el cual, afecta al sistema nervioso central, que tiene como consecuencia síntomas motores y no motores en pacientes que la padecen. Actualmente, para diagnosticar la enfermedad se realizan ciertos exámenes especializados como lo son la resonancia magnética y tomografías, que pueden llegar a ser altamente costosas e invasivas para los pacientes que se someten a este tipo de exámenes [7].

En el ámbito de cómo detectar esta enfermedad, el análisis de señales mediante la voz es una variante para la detección de esta, esto debido a las afectaciones que padecen los pacientes en su voz, las cuales mediante técnicas de procesamiento de señales y extracción de características se pueden identificar. Algunos cambios en la voz que se pueden presentar son frecuencia fundamental, temblores en la voz, reducción de su intensidad, etc. Esto anterior permitiendo que a través de modelos de aprendizaje basados en el reconocimiento de patrones se pueda realizar la detección de la enfermedad.

En este apartado de marco teórico se abordan los principales conceptos que están relacionados con la enfermedad de Parkinson, como la importancia de análisis de la voz en audios para la detección de esta y metodologías basadas en máquinas de aprendizaje automático para la clasificación de pacientes.

2.2.1 Enfermedad de Parkinson

De acuerdo con la Institucion Nacional de Desordenes Neurologicos y Ataques (NINDS) [15], la enfermedad de PD es un trastorno del movimiento que afecta directamente el sistema nervioso humano. Este trastorno genera células nerviosas que se debilitan y se provoca su muerte lo que puede provocar diferentes sintomas

como problemas de movimiento, temblores y afectaciones en el habla, entre otros. Este último es donde nos centramos en este proyecto.

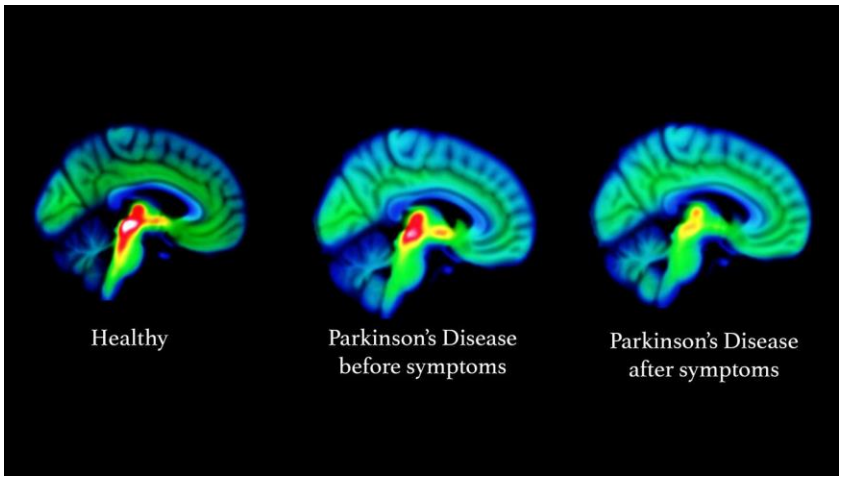

Figura 2-1 Imagen del cerebro humano en el camino de la perdida de serotonina debido al PD. Fuente [16]

La sintomatología de la enfermedad de PD aparece gradualmente y se incrementa con el tiempo. Aunque todos estamos expuestos al PD, se estima que los z tienden a ser más propensos a padecerlo que las mujeres. A pesar de esto, existen factores de riesgo más relevantes como la edad, ya que más del 90% de las personas con PD tienen más de 60 años, sin embargo, a las personas más jóvenes no están exentas de riesgo. Además, existen otros factores de riesgo como la predisposición hereditaria, además estudios afirman que las personas que viven en zonas rurales están expuestas a ciertos químicos que podría impactarlas y generar la enfermedad [15].

El PD afecta directamente al sistema nervioso y las afectaciones están directamente relacionadas al cerebro debido a que existen células nerviosas asociadas a la dopamina a la producción de dopamina, proceso en el cual estas células son insuficientes o se mueren, se produce menos cantidad de este neurotransmisor. Además, se pierde otro nervio llamado noradrenalina, el cual, es el principal mensajero del sistema nervioso simpático que controla varias funciones del cuerpo humano, como la frecuencia cardiaca y presión arterial, dando así, explicación a algunos síntomas de la enfermedad como la fatiga y ciertos cambios en el sistema cardiovascular [15].

2.2.1.1 Sintomatología

Existe una variedad de sintomatologías asociadas con el PD, que pueden manifestarse de diferentes maneras en los pacientes. Esto implica que no hay un

solo signo que permita predecir si alguien padece de la enfermedad. Además, el PD puede afectar inicialmente un lado del cuerpo, sin embargo, etapas avanzas pueden llegar a ser ambos. Dentro de los signos más comunes se encuentran [15]:

Temblor: El temblor en una extremidad, como la mano, es el síntoma más común dentro de las personas que padecen esta enfermedad, aunque este puede aparecer primero en el pie o mandíbula. Este trastorno es un movimiento de ida y venida en la extremidad afectada, cabe resaltar que es un movimiento involuntario del cuerpo y tiende a mejorar cuando las personas quieren realizar un movimiento específico o durante el sueño.

Existen otros tipos de síntomas, que también pueden afectar a los pacientes con PD como lo son:

- Inestabilidad en la postura
- Problemas de salud mental
- Problemas al masticar
- Problemas urinarios
- Problemas en la piel
- Dificultad para dormir
- Demencia
- Fatiga
- Disfunción eréctil

2.2.1.2 Problemas del habla a causa del Parkinson

A parte de estos problemas antes mencionados, también se encuentra la dificultad para hablar, que es el foco de este proyecto. La mayoría de los pacientes que padecen de la enfermedad presentan este síntoma, lo que los lleva a hablar de manera más lenta o monótona. Teniendo en cuenta esto, los pacientes generalmente presentan molestias al hablar, un deterioro que puede variar e incluir hipotonía, volumen monótono, o cambio en la velocidad al hablar, o disartria hipocinética. Estas afectaciones pueden tener implicaciones significativas en la vida diaria de las personas con PD, e incluso en los casos más severos se puede presentar problemas al masticar y tragar la comida [17].

A pesar de que las afectaciones pueden ser más severas, alrededor del 90% de las personas con PD presentan problemas de disartria, que son problemas del habla [7]. Solo entre el 10% y el 20% presentan los síntomas más severos llamados disartria hipercinética. Además, las personas pueden caer en problemas en el tono

de la voz, pues, estos pueden estar pensando que hablan en un volumen adecuado, cuando no es así, perdiendo la percepción de la voz y afectando la comunicación de los pacientes con PD con su entorno.

Las afectaciones vocales pueden hacerse notorias incluso en la etapa prodrómica de la enfermedad, antes de presentar cualquier movimiento motor involuntario provocado por el PD, así como en etapas iniciales. Sin embargo, esto no excluye a las personas que padecen de PD en etapas avanzadas [17].

2.2.2 Importancia detección temprana

Clínicamente, no se cuenta con un examen especializado, que permita determinar con seguridad la presencia de PD. De acuerdo con las afectaciones vocales, únicamente existe una evaluación cualitativa que puede hacer un profesional de la salud, que está basada en un subtema específico de la Escala Unificada de Calificación de la Enfermedad de Parkinson (UPDRS) [18].Teniendo en cuenta lo anterior, aun no se puede precisar una evaluación objetiva de si una persona cuenta con afectaciones en la voz debido al PD.

Además, si se considera los costos de manutención de pacientes con PD, en Estados Unidos son $11 billion USD anuales aproximadamente, donde la mayoría de las personas que absorben este valor son pacientes que presentan PD en etapas avanzadas. Por lo anterior, si se realiza una detección certera de la enfermedad en etapas tempranas, podría ser beneficioso económicamente, como para la calidad de vida de los pacientes [19]. Dentro de los exámenes asociados al PD encontramos:

- DatScan: Imágenes de la dopamina en el cerebro
- Resonancia magnética: Imagen del cerebro del paciente.
- Biomarcadores: Dos pruebas diferentes: Flujo espinal y biopsia de la piel.

2.3 Metodología

2.3.1 Enventanado a tiempo fijo

La normalización de audio a tipo fijo es importante, debido a la extracción de los coeficientes cepstrales en frecuencia (MFCC) o diferentes características, que requieren un proceso de segmentación por ventanas de tiempo fijo [20]. Otro beneficio es la optimización del procesamiento computacional, puesto que, el procesamiento audios de duraciones variables o duraciones variables, se requiere más tiempo de procesamiento y de memoria.

2.4 Machine learning

El *Machine Learning* (ML) es un tipo de inteligencia artificial basada en algoritmos que buscan la reducción de la asistencia humana para completar tareas con altos niveles de inteligencia, similar al funcionamiento del cerebro humano. Estos procesos de máquinas aprendizaje están inmersos en muchos contextos como diagnósticos clínicos, videojuegos, cuidado personal, entre otros [21]. Para llevar a cabo un proyecto con ML se debe tener en cuenta los siguientes pasos:

- Descripción del problema
- Base de datos
- Preprocesamiento de datos
- Modelos de ML
- Evaluación

2.4.1 Machine Learning en aplicaciones médicas

Las técnicas de ML cada vez son más utilizadas en este tipo de aplicaciones, sobre todo en el pronóstico de enfermedades, donde se emplean grandes cantidades de datos para detectar enfermedades como el cáncer, PD, entre otros. Gracias a estas técnicas, se ha logrado detectar tempranamente enfermedades de manera precisa, lo cual ha apoyado el diagnóstico y ahorrado tiempo en tratamientos, siendo muy beneficioso para este campo de aplicación [21].

2.4.2 Shallow Learning

Shallow learning (SL) es una rama del ML inspirada en el reconocimiento de patrones para procesos de clasificación de datos, de manera simple y eficaz, ideales para aplicaciones con recursos computacionales limitados o para aplicaciones poco complejas. SL necesita de asistencia humana para imponer la manera en la que se comportará la máquina, incluyendo configuración de hiperparámentros, imposición de datos, entre otros. Entre las máquinas de SL más comunes se encuentran Support Vector Machine (SVM), K-neigbhors (KNN), Decision tree (árbol de decisiones), entre otras [22].

2.4.2.1 Support Vector Machines (SVM)

Las máquinas de soporte vectorial (SVM) hace parte de las técnicas de aprendizaje supervisado, utilizada para la clasificación y regresión lineal. En este proceso, se busca encontrar la margen en hiperplanos que mejor exactitud de separación se obtenga entre clases para realizar una clasificación, donde se busca el hiperplano optimo que trabaje como frontera con el fin de minimizar el error y se maximizar la precisión. Esta máquina es útil para aplicaciones de datos linealmente separables, sin embargo se pueden manejar casos no lineales mediante kernels [23].

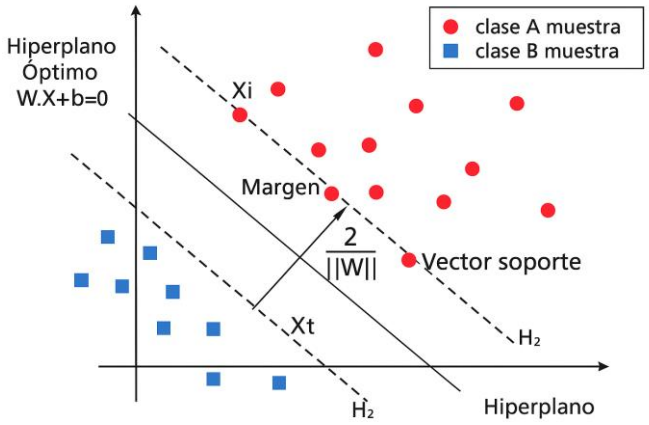

Figura 2-2 Diagrama de funcionamiento de SVM. Fuente: [24]

2.4.2.2 K-Nearest Neighbors (KNN)

El algoritmo KNN está basado en un modelo el cual clasifica teniendo en cuenta los ejemplos de entrenamiento que estén más cercanos a la prueba. Este algoritmo clasifica basado en la proximidad que se tenga con otros puntos de datos, empleando la distancia euclidiana como métrica de distancia. Para esto se realiza una votación entre los vecinos más cercanos, el punto que obtenga la mayoría de los votos en una clase, es asignado para la misma, como se puede observar en la Figura 2 - 2 con el punto P. Este algoritmo es simple y efectivo para tareas de clasificación con datos etiquetados, y el rendimiento puede ser afectado debido a la elección de número de vecinos [23].

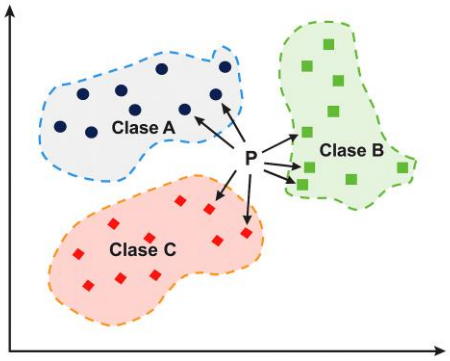

Figura 2-3 Diagrama de funcionamiento de KNN. Fuente: [25]

2.4.2.3 Random Forest (RF)

El clasificador RF está compuesto de numerosos arboles de decisión, cuyo funcionamiento se basa en la aleatorización, como se puede observar en la Figura 2 - 4, los nodos de los áboles obtienen etiquetas a través de estimaciones durante el proceso. Además, cada nodo tiene internamente una prueba que divide el espacio de los datos, mejorando así la clasificación. Por último, el proceso de clasificación mediante RF se realiza mediante una votación de los árboles de decisión que cada uno clasifica de manera independiente y la clase que obtiene la mayoría de los votos es la asignada a la clase específica [23].

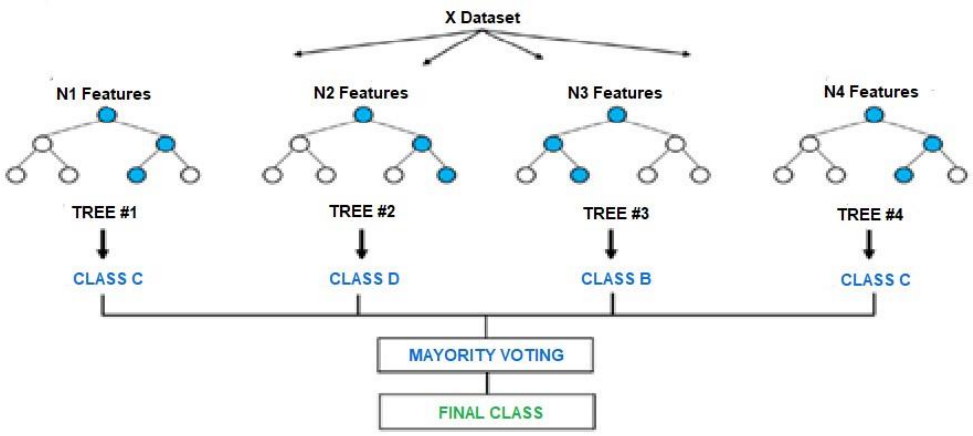

Figura 2-4. Diagrama de funcionamiento de RF. Fuente: [23]

2.4.2.4 Decision Tree (DT)

El clasificador DT es un diagrama con una estructura de árbol, el cual los nodos superiores representan las raíces y los nodos internos son las pruebas de características. Los bordes de este diagrama son los resultados de las pruebas, y el nodo al final significa a la clase que se asignó. Dentro de las claves que tiene este algoritmo es encontrar las características más determinantes en clasificación que accionen como raíz del árbol, que lo convierte en un clasificador altamente efectivo con pocos recursos computacionales [26].

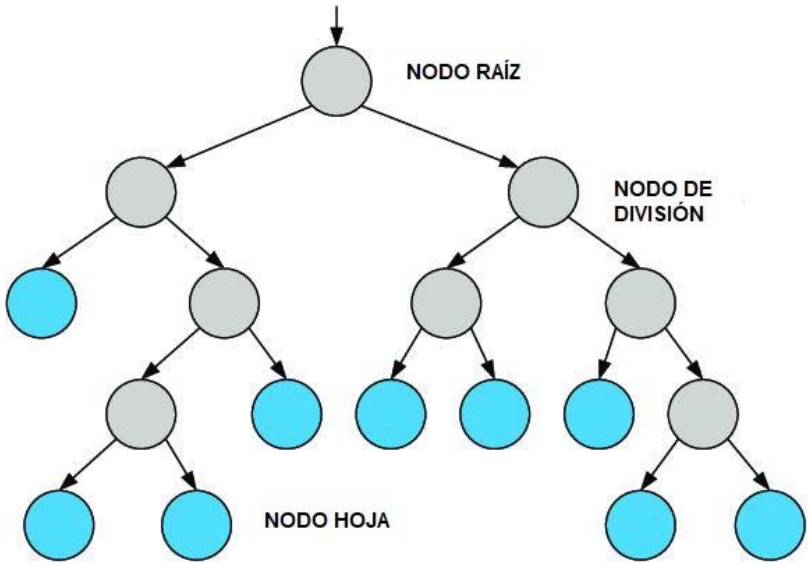

Figura 2-5 Diagrama de funcionamiento de DT. Fuente: [27]

2.4.3 Deep Learning (DL)

Deep learning (DL) que, a diferencia del SL, no requiere de una extracción manual de características ya que cuenta con la aprender de manera autónoma a partir de los datos, pudiendo identificar patrones más complejos, lo que hace que sea un tipo de ML poderoso para tareas de reconocimiento de imágenes, series temporales, entre otros. Esto se debe a que está compuesta por redes neuronales profundas las cuales utilizan grandes cantidades de datos y están diseñadas por muchas capas de neuronas, donde cada una de estas alimenta de manera positiva la máquina mejorando la precisión de clasificación. Existen diferentes tipos de máquinas de aprendizaje profundo entre ellas se encuentran, Multi-Layer Perceptron (MLP), Recursive Neural Networks (RNN), Convolutional Neural Networks (CNN), entre otras [22].

2.4.3.1 Convolutional Neural Network (CNN)

Las redes neuronales convolucionales (CNN) hacen parte de las arquitecturas de máquinas de aprendizaje profundo más utilizadas en diferentes aplicaciones, esto, debido a la capacidad de extraer características relevantes y reconocer patrones a partir de datos de entrada, sin la intervención humana para la selección de estas características. Están compuestas por capas como las convolucionales, de agrupamiento (*pooling*), funciones de activación, capas conectadas y funciones de pérdidas. Las capas convolucionales procesan la información, las capas de *pooling* reducen la dimensionalidad de las capas, y por otro lado las capas totalmente conectadas son las que redimensionan la información para ser procesada [22].

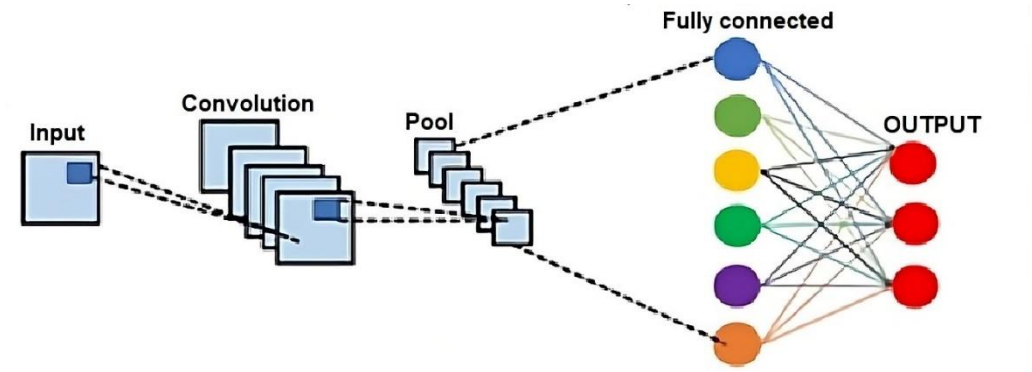

Figura 2-2-6 Diagrama de funcionamiento de CNN. Fuente: [28]

2.4.3.2 Resnet18

La principal característica de esta arquitectura son las conexiones residuales que evita problemas de desvanecimiento de gradiente y facilita el entrenamiento[29]. Estas conexiones están compuestas por dos tipos: Conexión tipo I: Utilizadas para cuando las dimensiones de entrada y salida son iguales. Conexión tipo II: Se utilizan cuando pasa el caso contrario que el mapa de características disminuye y el tamaño de los filtros aumenta [30].

2.4.3.3 EfficientNet-B0

Es una arquitectura de una red neuronal convolucional (CNN), la cual fue desarrollada Tan y V. Le[31]. Esta red presenta como objetivo principal alcanzar un equilibrio entre precisión y eficiencia, mediante el uso de un método llamado

escalado compuesto, que ajusta la profundidad en número de capas y el ancho de número de canales de manera simultánea [32].

2.5 Matriz de confusión

La matriz de confusión es una herramienta utilizada para evaluar el rendimiento de los modelos implementados. En ella se distinguen cuatro categorías fundamentales [12]:

- **True Positive (TP):** Representa los casos en los que el modelo identifica correctamente la presencia de la enfermedad en un paciente.
- **True Negative (TN):** Indica que el modelo clasifica correctamente a un paciente como no afectado por la enfermedad.
- **False Positive (FP):** Ocurre cuando el modelo diagnostica erróneamente la enfermedad en un paciente que, en realidad, no la padece.
- **False Negative (FN):** Se da cuando el modelo clasifica incorrectamente a un paciente enfermo como si estuviera sano.

Esta matriz permite analizar la precisión de un modelo y detectar posibles áreas de mejora en la clasificación de los datos.

Para medir el rendimiento de estos se tienen unas métricas dadas como:

$$Exactitud = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2 - 1)$$

$$Sensibilidad = \frac{TP}{TP+FN} \qquad (2 - 2)$$

$$Presición = \frac{TP}{TP+FP} \qquad (2 - 3)$$

$$F1 - Score = \frac{2*sensibilidad*presicion}{sensibilidad+presicion} \quad (2 - 4)$$

3. Metodología de la Solución Propuesta:

3.1 Introducción

En este capítulo se documentará detalladamente las soluciones propuestas de diseño de máquinas de aprendizaje superficial y profundo, con el fin de determinar la presencia o no presencia de PD en señales de audio. Se presentan los diagramas

de bloques que describen el flujo de las etapas empleadas en este proyecto para las dos metodologías, la primera de ellas para aprendizaje superficial y una segunda para aprendizaje profundo, que cuentan con ciertas similitudes, pero las dos encaminadas a la detección de presencia de PD en grabaciones de audio.

La primera propuesta para los cuatro modelos de aprendizaje superficial contiene la etapa de preprocesamiento, que está compuesta por subetapas como la eliminación de silencios y picos, filtrado de la señal, normalización de los audios, recorte de audios, submuestreo de frecuencia y aumento de datos. La segunda etapa se compone por la extracción de características, normalización y selección de estas. En tercera etapa se encuentran las máquinas de aprendizaje superficial implementadas, como KNN, SVM, DT y RF. Por último, se encuentra la clasificación dada. Para estas etapas se emplearon en el entorno de Google Colab, con el lenguaje de programación Python, que permite la implementación de máquinas de SL y DL, a partir de librerías como SKLearn y TensorFlow

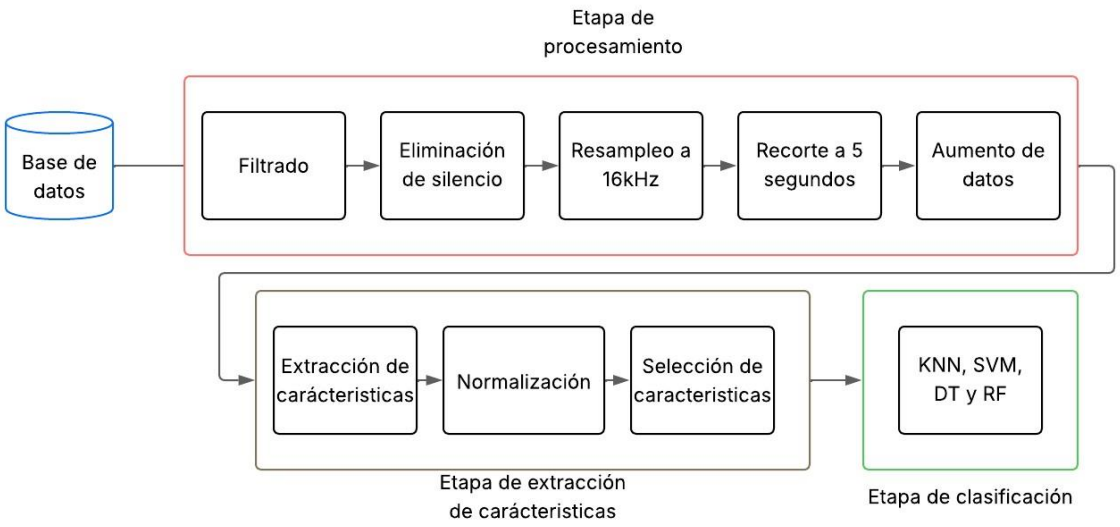

Figura 3-1 Diagrama de bloques implementado para máquinas SL.

Para la metodología con aprendizaje profundo, se presenta la misma etapa procesamiento de la metodología de aprendizaje superficial, ya que, en etapas de procesamiento se realizan los mismos procesos, mientras que la etapa de extracción de características se remplaza por la etapa de conversión a espectrogramas de los audios completos, esto con el fin de adecuar los datos de entrada para las máquinas de DL. Por último, está la clasificación de las máquinas CNN como ResNet18 y EfficientNet-B0.

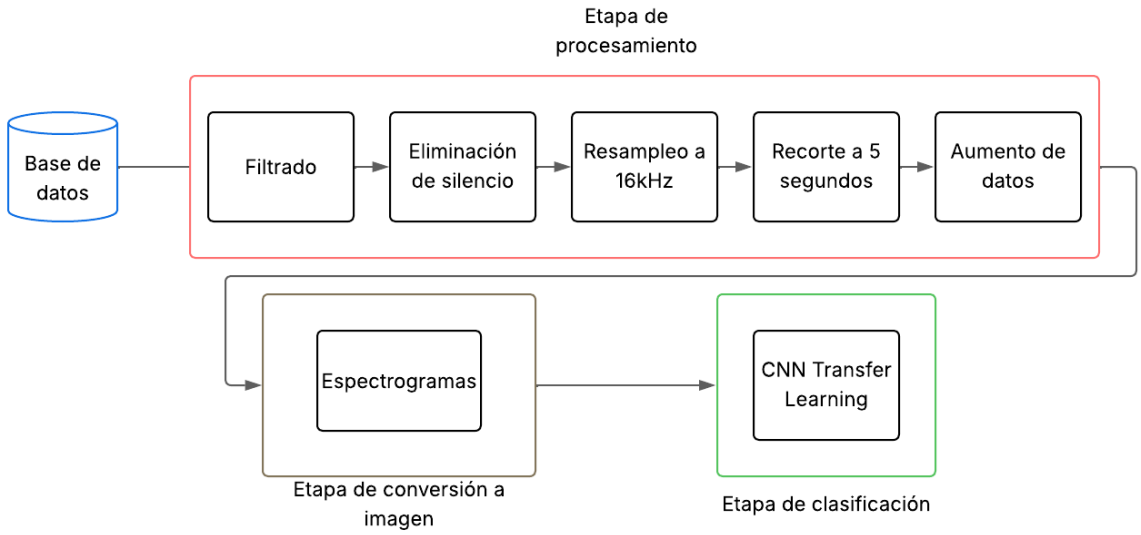

Figura 3-2 Diagrama de bloques implementado para máquinas DL.

3.2 Etapa de preprocesamiento

3.2.1 Base de datos

La elección de la base de datos es una tarea que resulta crucial en la fase inicial antes de realizar cualquier procedimiento. La base de datos que se elija afecta directa y significativamente el proceso, puesto que puede depender de la calidad de los audios en este caso los resultados que se obtengan posteriormente con las máquinas de aprendizaje. Por lo anterior, esta elección es muy importante, pues se debe escoger una base de datos que contenga una cantidad de audios considerable para realizar el proceso, que permita a su vez aplicar técnicas de preprocesamiento, extracción y clasificación de PD.

Para ello se tuvieron en cuenta tres bases de datos diferentes, la primera de ellas PC-GITA, que es una base de datos en español [30], conteniendo la entonación de la vocal “A”, en segundo lugar, se investigó acerca de MVDR-KCL, que son pacientes nativos en el idioma inglés leyendo una frase y realizando diálogo espontáneo [33]. Por último, la base de datos IPVS con las entonaciones de todas las vocales en italiano, además de silabas sostenidas. Los datos de estas bases de datos se encuentran en la siguiente tabla resumen:

Tabla 3-1 Tabla resumen de las bases de datos consultadas.

Base de datos	Idioma	Tipo de grabación	Número de pacientes	Número de muestras
PC-Gita	Español	Vocal “A”	50	100
MVDR-KLC	Inglés	Frase y diálogo espontáneo	37	74
IPVS	Italiano	Vocales y sílabas	47	658

Como se muestra en la Tabla 3 – 1, la base de datos PC-GITA cuenta con 50 pacientes y únicamente 100 muestras de la vocal “A”, que lo hace una base de datos con pocas muestras, por otro lado, se presenta MVDR-KLC con una frase leída y diálogo espontáneo entre el paciente y un entrevistador, se podría decir que es una base de datos completa por el tamaño de los audios de dos minutos para los diálogos, sin embargo, tendría la limitación de que es un idioma. Por último, se presenta la base de datos IPVS, con 47 pacientes, pero con 658 grabaciones que daría un total de 14 grabaciones de pacientes entre vocales y silabas, lo que lo hace una base de datos más robusta.

3.2.1.1 Italian Parkinson´s Voice and Speech Dataset (IPVS)

Esta base de datos incluye grabaciones de audio de 50 personas, de las cuales 28 de ellas padecen de PD y las restantes 22 son controles sanos. Esta se ha seleccionado ya que cuenta con 14 audios por persona, los cuales de dividen de la siguiente manera:

- Dos audios leyendo un párrafo asignado.
- Un audio entonando la sílaba “ka” por cinco segundos.
- Un audio entonando la sílaba “pa” por cinco segundos.
- Dos audios entonando la vocal “a”.
- Dos audios entonando la vocal “e”.
- Dos audios entonando la vocal “i”.
- Dos audios entonando la vocal “o”.
- Dos audios entonando la vocal “u”.

Estas se realizaron tomando cada audio con treinta segundos de separación entre ellos, esto, con el objetivo de que el paciente pudiera tomar el aire suficiente y fuera una toma certera de un momento del paciente en reposo. Además, se precisa que los datos fueron tomados con el micrófono de 15 cm a 25 cm de la boca de las personas, así como la temperatura del aula que fue aproximadamente a 22°C en diferentes días[34].

Tabla 3-2. Tabla de distribución de los pacientes que fueron grabados para la base de datos IPVS

Categoría	Pacientes con PD	Pacientes sanos
Número de pacientes	28 (19 hombres, 9 mujeres)	22 (13 hombres, 9 mujeres)
Edad aproximada	60 – 77 años	40 – 80 años
Idioma	Italiano	
Frecuencia de muestreo	16 kHz, algunos 48 kHz	

Para este proyecto únicamente serán considerados los audios que contienen las vocales sostenidas, excluyendo los audios de discurso y de sílabas sostenidas, esto debido a la similitud que se puede presentar con los idiomas provenientes del latín e idiomas romances como el español, francés, portugués, etc [35], lo cual puede ser beneficioso si se pretendiera interpolar este proyecto a un idioma romance diferente al italiano. Además de ello, se puede evidenciar que hay una cantidad considerable de audios a pesar del recorte, terminando con 470 audios de diferentes duraciones, lo que resulta beneficioso para el proceso de clasificación e impactará de manera positiva en los entrenamientos de las máquinas a implementarse.

Esta base de datos además fue considerada gracias a la calidad de los audios, que fueron grabados en un ambiente de silencio, en una habitación sin eco, con un mismo micrófono para todos los pacientes a una distancia de 15 a 25 cm de estos, teniendo en cuenta un descanso entre audios. Por otro lado, se tiene en cuenta la accesibilidad de la base de datos, la cual se puede descargar gratuitamente de la página de IEEE[34], así como su uso en diferentes proyectos de investigación[13][9].

3.2.2 Preprocesamiento de los audios

El preprocesamiento de la base de datos de audio, presenta como objetivo el mejorar la calidad de los audios antes de realizar la extracción de características [36]. Para esto, se busca evitar sonidos o interferencias no deseadas en los audios para que las máquinas a implementarse puedan reconocer patrones de la manera deseada. Dentro de las técnicas implementadas se encuentran, la etapa de filtrado

audio para eliminar frecuencias no deseadas o no relevantes, así como unificar la amplitud para reducir diferencias entre pacientes, eliminación de silencios, recorte de los audios a una misma duración y remuestreo de audio a 16 kHz para obtener estos lo más homogéneos posibles para la posterior extracción de características.

3.2.2.1 Filtrado

Para realizar el proceso de filtrado en procesamiento de señales, usualmente se aplica un filtro pasa-bajas con el objetivo de suprimir frecuencias no deseadas para el análisis que se está implementando. Con respecto a este caso específico se implementó un filtro pasa-bajas con una frecuencia de corte en 3 kHz, esto por medio de un filtro Butterworth para eliminar las componentes frecuenciales no relevantes en este proyecto, teniendo en cuenta la Figura 3 - 5, donde se evidencia que la mayor parte de la información se encuentra por debajo de la frecuencia de corte deseada.

En el proceso de filtrado se aplica el filtro Butterworth de cuarto orden por medio de la función *FiltFilt* y *Butter* de Python con hiperparámetros de frecuencia de corte a 3 kHz, tipo de filtro paso bajo, orden 4 y frecuencia de muestreo del audio. Este filtro es seleccionado debido a una de las ventajas que proporciona distorsión mínima en la señal, que lo convierte relevante para trabajar en señales de voz [37]. Se establece una frecuencia de corte en 3 kHz, es decir, las componentes superiores a este valor van a ser suprimidas suavemente, puesto que las frecuencias más relevantes para esta aplicación en específico son inferiores a 3 kHz como en el estudio [38] donde se realiza un estudio de miles de audios grabados, donde se estudia la frecuencia fundamental de la voz en los mismos obteniendo resultados de menos de 300 Hz. Teniendo en cuenta esto eliminar las frecuencias superiores a 3 kHz y la Figura 3 - 3, se reduce el ruido de los audios y se mejora la calidad de los audios para la posterior extracción de características.

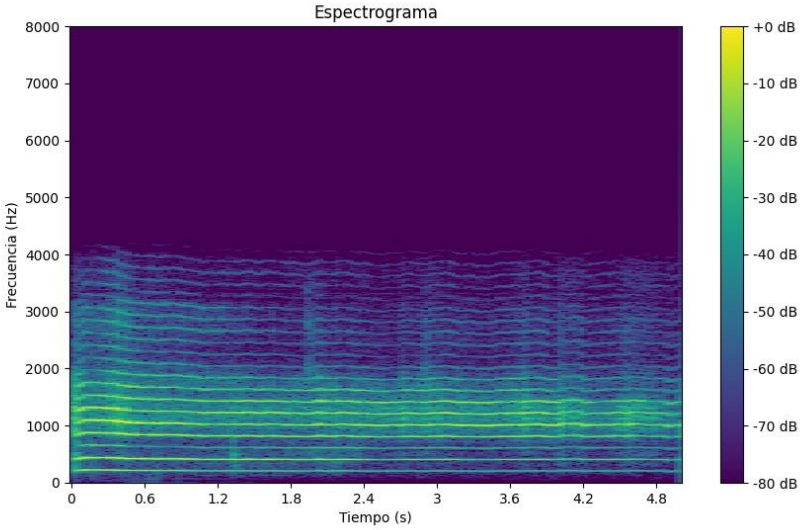

Figura 3-3 Espectrograma de un audio sano de la base de datos luego de haber sido filtrado. Fuente: Propia.

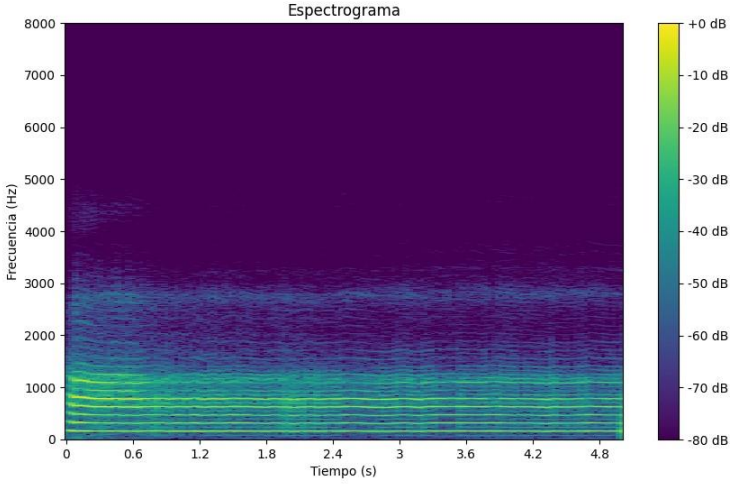

Figura 3-4 Espectrograma de un audio con Parkinson de la base de datos luego de haber sido filtrado. Fuente: Propia.

3.2.2.2 Eliminación de offset

La presencia de offset en señales de audio hace referencia a un desplazamiento que se presenta a lo largo de la señal, con respecto a la línea de base cero, las principales causas de esto son el ruido de fondo, algunos errores que se pueden presentar durante la grabación de audio o en algunos casos el micrófono con el cual se realizan las grabaciones. Presentar offset dentro de una señal afecta negativamente, pues, se puede presentar un análisis con sesgado en amplitud y añade ruido en la etapa de extracción de características, lo que puede producir valores erróneos extraídos.

El proceso realizado para eliminar el offset es una técnica de centralización de la señal de audio, donde se calcula la media de la amplitud que se tiene en cada audio y posteriormente restarlo en cada muestra del audio, como se puede observar en la siguiente fórmula:

$$X_c(n) = x(n) - \mu \qquad (3 - 1)$$

Donde $x(n)$ representaría la muestra de la señal y μ la media calculada, dando resultado a $X_c(n)$ que sería el audio resultante centralizado. Este proceso brinda una señal con una media muy cercana a cero, lo que implica mejoras en la etapa de extracción de características, que está directamente relacionada con los modelos que no tendrían una variabilidad de amplitud debido a este procedimiento realizado.

3.2.2.3 Eliminación de silencios

En la base de datos resulta fundamental la eliminación de silencios, ya que, para la detección de PD, estos silencios prolongados o espacios donde el tono de voz es muy débil por diferentes motivos, puede introducir información irrelevante al momento de la extracción de características y afectar a los modelos implementados, puesto que se estaría incluyendo estos espacios dentro de nuestro análisis.

Dentro de la base de datos se identificaron numerosos audios, los cuales presentaban silencios al inicio o incluso en medio de los audios, ya que, puede

suceder que los pacientes se alejen del micrófono y presentar una amplitud de la señal muy baja, que afectaría directamente en las etapas posteriores, el promedio de los audios antes del recorte fue de 10.88 segundos.

En los audios pueden existir silencios al inicio, hasta incluso presentar silencio al final, para corregir esto se implementó una estrategia de segmentar la señal en intervalos fijos de 12.5 ms y realizar evaluación de energía en cada uno de estos. Para realizar el proceso, se calcula la raíz cuadrada media (RMS), la cual mide la energía del segmento y se compara con un umbral dado del 10% de la amplitud pico, permitiendo determinar si se conserva el segmento o se descarta.

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x(n)^2} \qquad (3 - 2)$$

Teniendo en cuenta lo anterior de esta forma la reducción de los audios a procesar fue de promedio 0.92 segundos, esta eliminación de silencios y espacios innecesarios en las señales, pudieron permitir una mejora considerable para posteriores etapas del proyecto, además de esto, permite estandarizar los audios a ciertas condiciones dadas y aproximar la base de datos a ser homogénea

3.2.2.4 Remuestreo a 16k Hz

En el contexto de este proyecto acerca de detección de PD, tener la frecuencia de muestreo a 16 kHz, se basa en que primeramente la mayor parte de información útil para nuestra aplicación puede encontrarse entre 100 Hz a 3500 Hz[39], además, si se sigue el teorema de Nyquist, donde la frecuencia de muestreo debe ser al menos del doble de la frecuencia máxima de interés [40], que si visualizamos la Figura 3 - 5 donde esta se encuentra cerca de 3 kHz, estaríamos cumpliendo perfectamente este teorema, por lo cual un muestreo a 16 kHz vendría siendo suficiente para capturar con precisión la información de los audios dentro de la base de datos.

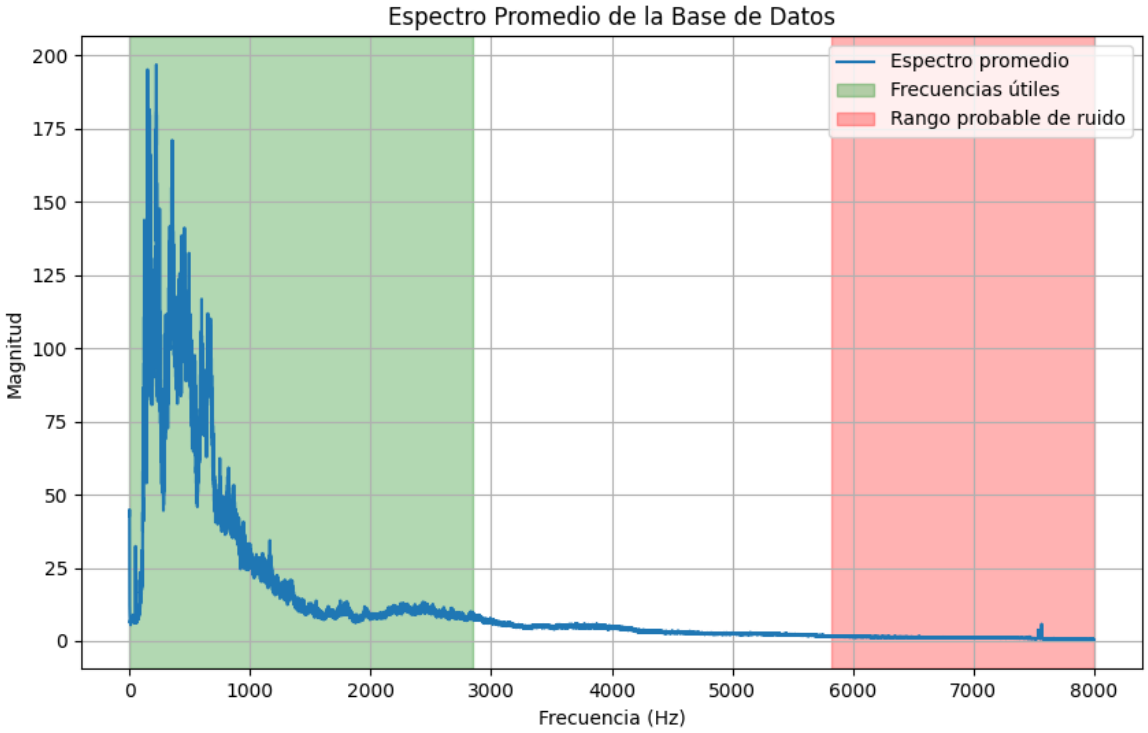

Figura 3-5 Espectro promedio de la base de datos. Fuente: Propia

Al haber realizado la verificación de las muestras por segundo que presentaban los audios, se encontró que algunos de estos estaban grabados a 48 kHz, esto, a pesar de que en [13] se precisaba que todos los audios estaban grabados a 16k Hz. Por lo anterior, se realizó un remuestreo general de los audios a 16k Hz, que trae consigo beneficios como lo son la homogeneidad en los datos, que es fundamental para la extracción de características y para los algoritmos de clasificación, pues en caso de presentarse datos con diferentes frecuencias de muestreo, pueden encontrarse inconsistencias en clasificación.

Otro aspecto para tener en cuenta es la optimización que se presenta en el almacenamiento, puesto que, al reducir la frecuencia de muestreo de 48 kHz a 16 kHz, se estaría representando la misma información con una cantidad menor.

3.2.2.5 Enventanado de audio a 5 segundos

Para realizar el proceso de extracción de características, se pretende tener audios normalizados en cuanto a duración, debido a que se espera un proceso de segmentación por ventanas de tipo rectangular para la extracción. Por lo anterior,

establecer una longitud fija que permite que los audios tengan homogeneidad, una mejora en la compatibilidad entre los mismos.

Por estos motivos anteriormente expuestos, como además la verificación de la longitud de los audios luego de haber realizado el proceso de eliminación de silencios, se evidencia que todos los audios están por encima de los 5 segundos de duración, por lo que no se perderían audios si se recortan a esta longitud, además cabe resaltar que los audios tuvieron un promedio de 9.92 segundos antes de haber sido recortados.

3.2.2.6 Aumento de datos

En esta subetapa del preprocesamiento de los audios, se toman las grabaciones recortadas y se realiza proceso de aumento de datos mediante la separación mediante ventanas [41], la cual tiene como fin aumentar el número de muestras a partir de los datos ya existentes. Para llevar a cabo este proceso se definió un tamaño de las ventanas, como también el solapamiento (*overlapping*) entre las mismas.

Los audios se segmentaron en ventanas de 4096 muestras con un solapamiento del 50% [42] que implica 2048 muestras, proceso el cual ayuda a conservar la continuidad de la señal y evitar la pérdida de información relevante de la ventana. Dado que los audios tienen una duración de 5 segundos y los audios están muestreados da 16 kHz se tiene la ecuación:

$$5s \ x \ 16.000 \ sps = 80.000 \ sps \qquad (3-3)$$

Posteriormente las ventanas que se han divido en 4096 muestras son en el tiempo:

$$\frac{4096 \ sp}{16.000 \ sps} = 256 \ ms \quad (3-4)$$

Para el número de segmentos obtenidos se hace el siguiente cálculo debido a que el overlapping es de 2048 muestras:

$$\frac{80.000 \ sps - 4096 \ sps}{2048 \ sps} + 1 \approx 38 \ muestras \ (3 - 5)$$

Al haber aplicado esta técnica de aumento de datos a todas las grabaciones de audio, finalmente se pasa de tener 457 audios a tener 17.357 ventanas de audio para su uso en extracción de características y audio.

3.3 Extracción de características

Para el proceso de extracción de características, la cual es una etapa esencial en este proyecto para capturar la información proveniente de los audios. Se calculan 51 características de diferentes tipos como el temporal, frecuencial, cepstral y la calidad vocal, teniendo en cuenta los antecedentes sobre qué características podrían ser más determinantes para este proyecto en específico, además debido al gran tamaño de nuestras características se implementó una etapa adicional para determinar las más discriminantes en este caso.

3.3.1 Características

En la etapa de extracción de características se consideraron tres dominios: temporal, frecuencial y acústico. El propósito de la selección realizada es abordar de forma integral la problemática de alteraciones que pueden sufrir los pacientes a raíz de la enfermedad, obteniendo una diversidad de características y posiblemente aumentando generalidad en clasificación.

Primeramente, se tiene el dominio temporal, donde se puede observar aspectos relevantes de la señal como ZCR, energía RMS, amplitud promedio, amplitud máxima, entropía de la señal, asimetría y curtosis. Estas características han sido relevantes en la clasificación de PD, puesto que se puede identificar la disminución vocal en pacientes, que es una afectación común para los mismos. Dentro de este proyecto se tiene amplitud media y máxima, energía RMS, entropía de la señal y la tasa de cruces por cero. En proyectos previos como [14] se implementaron este tipo de características para la clasificación de PD.

En segundo lugar, se presentan las características espectrales, donde en proyectos previos como [5][7][13][14], entre otros, donde se hacen uso de estas características como centroide espectral, ancho de banda espectral, planitud espectral, contraste espectral, caída espectral, frecuencia fundamental y densidad espectral..

En cuanto a las características cepstrales, se caracterizan de los audios los coeficientes de escala Mel (MFCCs), junto a las primeras y segundas derivadas

temporales. Varios proyectos como [5][13][14], entre otros antecedentes se emplearon estas características efectivas para métodos de clasificación de PD.

Por último, se adicionaron características acústicas como lo son Jitter, Shimmer y relación de armónicos-ruido. Estas características resultan relevantes, ya que como se menciona anteriormente el PD presenta afectaciones tales como disfonía, rigidez y pérdida de control en la voz. En varias investigaciones como [7][13], entre otros.

En conjunto estos cuatro grupos de características permiten una clasificación multidimensional de la voz, ya que se extraen diferentes tipos de características que alimentan positivamente los modelos de aprendizaje superficial, donde se busca discriminar pacientes con PD y pacientes sanos. Las características extraídas se extrajeron mediante librerías de Python como *Librosa, Numpy y Pandas*.

Tabla 3-3 Características extraídas mediante Python para las máquinas de aprendizaje superficial

Tipo	Característica	Descripción
Temporal	ZCR	Tasa de cruces por cero
	RMS energía	Energía media cuadrática
	Amplitud promedio	Valor promedio de la señal
	Amplitud máxima	Valor pico más alto alcanzado
	Entropía de la señal	Medida del desorden de la señal
	Asimetría	Indica si la señal está sesgada
	Curtosis	Forma de distribución de datos
Espectrales	Centroide espectral	Representación espectral peso relativo
	Ancho de banda espectral	Rango de frecuencias presentes
	Planitud espectral	Mide cuanto ruido hay presente
	Contraste espectral	Diferencia entre picos y valles de espectro
	Caída espectral	Porcentaje espectral que se encuentra debajo de umbral.
	Frecuencia fundamental	Frecuencia más baja y base del habla
	Densidad espectral	Distribución de la energía
Cepstrales	MFCC 1-12	Mel - Frecuencia cepstral
	Delta MFCC 1-12	
	DeltaDelta MFCC 1-12	
Acústicos	Jitter	Variaciones en frecuencia
	Shimmer	Variaciones en amplitud
	HNR	Relación armonios ruido
	NHR	Relación ruido armónico

Tabla 3-4 Fórmulas de las características extraídas

Característica	Fórmula
ZCR	$ZCR = \frac{1}{N-1}\sum_{n=1}^{N-1} 1[s(n).s(n-1) < 0]$
RMS	$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} s(n)^2}$
Amplitud media	$A_{mean} = \frac{1}{N}\sum_{n=1}^{N} abs(s(n))$
Amplitud Máxima	
Entropy	$H = -\sum_{i=1}^{M} P_i \log_2(P_i)$
Asimetría	$Skewness = \frac{\frac{1}{N}\sum_{n=1}^{N}(s(n)-\mu)^3}{\sigma^3}$
Curtosis	$Kurtosis = \frac{\frac{1}{N}\sum_{n=1}^{N}(s(n)-\mu)^4}{\sigma^4}$
Centroide espectral	$C = \frac{\sum_k f_k(asb(X(k)))}{\sum_k(asb(X(k)))}$
Ancho de banda espectral	$BW_P(t) = (\sum_k S[k,t](f_k - C)^2)^{\frac{1}{p}}$
Contraste espectral	$SC = \log(\frac{1}{\sigma N_k}\sum_{i=1}^{\sigma N_k} x'_{k,i}) - \log(\frac{1}{\sigma N_k}\sum_{i=1}^{\sigma N_k} x'_{k,N_k-i+1})$
Planitud espectral	$SFM = \frac{(\coprod_{k=1}^{K} a(k))^{\frac{1}{K}}}{\frac{1}{K}\sum_{k=1}^{K} a(k)}$
Caída espectral	$\sum_0^{fc} a^2(f) = 0.85\sum_0^{sr/2} a^2(f)$
Frecuencia fundamental	$f_0 = \frac{1}{t_0}$

MFCC 1-12	$MFCC_m = \sum_{m=0}^{M} S(m)Cos(\pi n(m+1/2)/M)$
Delta MFCC	$\Delta MFCC_m$
Delta Delta MFCC	$\Delta\Delta MFCC_m$
Jitter	$Jitter = \frac{1}{N-1}\sum_{i=1}^{N-1} abs(T_i - T_{i-1})$
Shimmer	$shimmer = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1} abs(A_i - A_{i+1})}{\frac{1}{N}\sum_{i=1}^{N} A_i}$
HNR	$HNR = 10\log_{10}\left(\frac{p_{harm}}{p_{noise}}\right)$
NHR	NHR $= \frac{p_{harm}}{p_{noise}}$

3.3.2 Estandarización

El cálculo de características requiere de un procedimiento de normalización, el cual tiene como fin evitar el sesgo de diferencia de escalas de las variables y afectar positivamente a los modelos de entrenamiento de este proyecto, mejorando la convergencia de este. Para realizar este proceso, se sometieron las características a la función de Python *MinMaxScaler* del submódulo *preprocessing* de la librería *SKLearn*, donde cada característica extraída cambia en su rango obteniendo finalmente una escala seleccionada entre 0-1 [43]. Este proceso fue realizado tanto para los datos de entrenamiento de los modelos como para los datos de evaluación por medio del método *fit* de Python.

3.3.3 Selección de características

El proceso de la selección de características tiene como objetivo es reducir la dimensionalidad de los datos, evadiendo el exceso de costo computacional manteniendo resultados con buen rendimiento en los clasificadores de máquinas de aprendizaje superficial, por otro lado, evitando el sobre entrenamiento que se podría presentar en estos algoritmos. Por estas razones se emplearon técnicas de selección de características, donde se elijan una cantidad de características que sean determinantes para el clasificador y con esta cantidad sacar los mejores resultados en clasificación.

En el actual proyecto se optó por utilizar un método llamado *Wrappers,* donde se se evalua el desempeño de todas las posibles combinaciones de características, seleccionando la combinación con mejor desempeño [44].

Además de realizar la selección de características mediante este método, se realizó un ajuste de hiper-parámetros explorando todas las combinaciones posibles de una cuadrícula de valores (GridSearch) y evaluando el desempeño del modelo en cada iteración, esto, con el fin de encontrar el mejor desempeño con diferentes números de características: 5, 7, 10.

3.4 Etapa de clasificación

En cuanto a las máquinas de SL se escogieron DTREE, SVM, KNN, y RF, de acuerdo con los antecedes consultados [7], [9], [12], [13], [13] . Mientras que para las máquinas de DL se escogieron dos CNN con diferentes arquitecturas de base. Para ambos tipos de máquinas se definieron los parámetros que se evaluarán.

3.4.1 Máquinas Shadow Learning

Para implementar las máquinas de SL, es necesario tener datos de entrenamiento y evaluación, a partir de las características numéricas extraídas de las grabaciones de audio. Teniendo en cuenta esto en el siguiente apartado se explicarán a profundidad las máquinas de aprendizaje a implementar anteriormente mencionadas, así como los parámetros a seleccionar mediante *GridSearch.*

3.4.1.1 Decision Tree

En cuanto al primer algoritmo de clasificación de Decision Tree, fue implementado mediante la librería *SKLearn*, donde se permiten ajustar diferentes parámetros, para el caso en específico se definieron los siguientes hiperparámetros de la función de la máquina de aprendizaje que son: función de evaluación "*Criterion*", profundidad máxima, mínimo de muestras para dividir y mínimo de muestras en hojas. Los posibles hiperparámetros a seleccionar se encuentran en la Tabla 3 - 5.

Tabla 3-5 Posibles hiperparámetros de Decision Tree (DT)para seleccionar mediante GridSearch

Hiperparámetro	Valores por evaluar
Criterio	“gini”, “entropy”
Profundidad máxima	“none”, “5”, ”10”, “20”
Mínimo de muestras a dividir	“2”, “5”, “10”
Mínimo de muestras en hojas	“1”, “2”, “4”

El primero de estos parámetros *Criterio* de decisión define la función para la medición de calidad de una división. Las opciones para tomar en cuenta son “gini” y “entropy”. El parámetro Gini realiza una medición de qué tan mezcladas están las clases dentro de un grupo en específico, por otro lado, *Entropy* calcula ganancia de información.

El criterio de *profundidad máxima* establece como su nombre lo dice la profundidad del árbol de decisión. Para este criterio un valor bajo evita un sobreajuste limitando la complejidad del modelo a implementar, mientras que si se selecciona un valor mayor se permite clasificar procesos más complejos. Con el valor de “None”, el árbol crecerá lo justo hasta que no queden más muestras a dividir.

El *mínimo de muestras a dividir* hace referencia al mínimo de muestras que se requieren para dividir un nodo. Un valor bajo de muestras con respecto al establecido hace que el nodo proceda a ser una hoja, mientras que valores mayores permiten generalización en el modelo, evitando divisiones poco determinantes.

Mínimo de muestras en hojas determina el número de muestras que componen un nodo de hoja. Este parámetro permite obtener árboles balanceados, asegurando la no creación de nodos pequeños, permitiendo, tener hojas más representativas.

Teniendo en cuenta lo anterior, con estos datos a evaluar, se buscará el modelo que mejor rendimiento obtenga en entrenamiento de la máquina, sistema que se empleó es el que está descrito en la Figura 3 - 1, con la única diferencia que en el apartado del modelo se encuentra esta máquina de SL DT [45].

3.4.1.2 Support Vector Machine

Para el algoritmo de clasificación de SVM, se implementó el ajuste de diferentes parámetros. Para el caso en específico se definieron los siguientes hiperparámetros para el funcionamiento de la máquina de aprendizaje que son: Parámetro de regularización “C”, tipo de kernel y el parámetro gamma. Las posibles hiperparámetros a seleccionar se encuentran en la Tabla 3 - 6.

Tabla 3-6 Posibles hiperparámetros de Support Vector Machine (SVM) para seleccionar mediante GridSearch

Hiperparámetro	Valores por evaluar
C	“0.1”, “0.5”, “1”, “3”, “5”, “10”
Kernel	“Linear”, “rbf”, “poly”, “sigmoid”
Gamma	“scale”, “auto”

El primer parámetro *C* es el encargado de realizar el control de separación de márgenes entre clases. Un valor bajo de este parámetro permite un margen más alto; sin embargo, se pueden presentar errores, mientras que, para un valor alto, intenta clasificar todos los puntos de entrenamiento, que puede hacer caer en un sobreajuste la máquina.

En cuanto al parámetro *kernel*, el cual está encargado de la función para la transformación de los datos, permitiendo así, encontrar diferentes hiperplanos que separen clases, sin embargo, este al ser un modelo biclase, funciona como el proceso para optimizar y encontrar el hiperplano con mejor rendimiento.

Por último, se cuenta con *gamma*, el cual realiza el control del modelo a entrenar, un valor alto de este hiperparámetro puede provocar sobreajuste en la máquina, mientras que, un valor bajo generaliza, permitiendo relaciones sólidas entre puntos a clasificar.

Teniendo en cuenta lo anterior, con estos datos a evaluar, se buscará el modelo que mejor rendimiento obtenga en entrenamiento de la máquina, el sistema que se empleó es el que está descrito en la Figura 3 - 1, con la única diferencia que en el apartado del modelo se encuentra esta máquina de SVM [45].

3.4.1.3 K-Nearest Neighbors

En el algoritmo de clasificación de KNN, se definieron los siguientes hiperparámetros para el funcionamiento de la máquina de aprendizaje: Número de vecinos, ponderación de los vecinos y métrica de distancia. Las posibles hiperparámetros a seleccionar se encuentran en la Tabla 3 - 7.

Tabla 3-7 Posibles hiperparámetros de K- Nearest Neighbors (KNN) para seleccionar mediante GridSearch

Hiperparámetro	Valores por evaluar
Número de vecinos	“3”, “5”, “7”, “9”
Ponderación de vecinos	“Uniform”, “Distance”
Métrica de distancia	“euclidean”, “manhattan”, “minkowski”

El *número de vecinos* tiene como función determinar el número de vecinos para tener en cuenta al momento de clasificar PD o HC. Un número pequeño de este parámetro puede afectar el modelo, puesto que sería más sensible a variaciones y propenso a sobreajuste. Por otro lado, con un valor un poco mayor, haría que se suavice la frontera de decisión, que podría provocar una clasificación más robusta.

En cuanto a la *ponderación de vecinos*, este determina la contribución para clasificación de cada vecino. En primer lugar “uniform” haría referencia a darle la misma importancia a cada vecino, mientras que “Distance”, haría que predomine los vecinos que estén más cerca al punto de clasificación, que podría afectar positivamente al modelo cuando los datos sean tan variables.

Por último, la *métrica de distancia* especifica la manera de calcular la distancia entre los puntos y los vecinos. “Euclidean” mide distancia recta en el espacio, mientras que, “Manhattan”, mide la distancia a partir de ejes ortogonales y, por último, “Minkowski” permite ajustar la distancia a implementar [45].

3.4.1.4 Random Forest

En el algoritmo de clasificación de Random Forest, se ajustan diferentes parámetros, en este caso en específico se definieron los siguientes hiperparámetros para el funcionamiento de la máquina de aprendizaje que son: Número de árboles, profundidad máxima, muestras mínimas para dividir un nodo, muestras mínimas en una hoja y muestreo con reemplazo. Las posibles hiperparámetros a seleccionar se encuentran en la Tabla 3 - 8.

Tabla 3-8 Posibles hiperparámetros de Random Forest (RF) para seleccionar mediante GridSearch

Hiperparámetro	Valores por evaluar
Número de árboles	“100”, “200”, “300”
Profundidad máxima	“none”, ”10”, “20”
Muestras mínimas para dividir un nodo	“2”, “5”, “10”
Muestras mínimas en una hoja	“1”, “2”, “4”
Muestreo con reemplazo	“True”, “False”

El *número de árboles* determina cuántos arboles de manera individual conformaran el bosque. Si se aumenta este número, se pretende mejorar el rendimiento del algoritmo, ya que se generaliza más el modelo, sin embargo, al hacer esto se podría afectar el costo computacional, por lo que es importante un equilibrio.

En *profundidad máxima*, se define la profundidad que alcanzarán los árboles. Al limitar este hiperparámetro, se busca controlar sobreajustes, puesto que los árboles podrían memorizar datos de entrenamiento, por lo que es importante configurar un valor determinado con el fin de tener un modelo más general.

Las *muestras mínimas para dividir un nodo* determinan el número de estas que se deben tener en un nodo para dividir ramas adicionales. Presentar un valor bajo, permitiría solucionar patrones complejos, por otro lado, para valores elevados, haría que se tenga un modelo más simple, forzando al modelo a dividirse cuando se tenga la cantidad suficiente de datos.

En cuanto las *muestras mínimas en una hoja* especifican el número de estas que deben encontrarse en una hoja. Este hiperparámetro se asegura de que cada predicción del modelo esté basada en suficientes datos, evitando el sobreajuste.

El muestreo con reemplazo determina si un conjunto de datos empleado debe ser una muestra obtenida como reemplazo. Al activar el *bootsrtap*, obtenemos variación en los árboles, mientras que, si este no está activo, los árboles se entrenarían sobre ellos mismos [45].

3.4.1.5 Importancia de características

Este método (Feature Importance) es implementado mediante Random Forest, el cual estima la importancia de cada característica para clasificación de PD. Como RF es un algoritmo basado en árboles de decisión, él mismo permite determinar qué tan importante es cada variable para la división de los datos. Al momento de que una característica se emplea para la división de un nodo, se calcula una impureza, este es acumulable, y al final se promedia con respecto al número de características [46].

Los resultados de la importancia de cada característica, donde se seleccionan las variables más discriminantes, eliminando las que no aportan al modelo, o que provocan un sobreajuste en este. Los beneficios de este proceso es la reducción del costo computacional y sobreajuste. Es importante realizar este tipo de procedimientos, puesto que se cuentan con 51 características, y se puede reducir la dimensionalidad, sin afectar considerablemente el rendimiento de la máquina [46].

3.4.2 Máquinas de aprendizaje profundo

Para implementar las máquinas de DL, se tuvieron en cuenta espectrogramas de las grabaciones de audio como datos de entrada, como tener dividido los datos de entrenamiento con un 70% y testeo 30%. Teniendo en cuenta esto, los espectrogramas requieren de un etiquetado, puesto que para realizar la clasificación los datos deben estar debidamente separados en entrenamiento y testeo, así como en pacientes con PD y pacientes HC. Para esto se hace uso de la librería *Torch* de Python, donde se contienen máquinas de aprendizaje profundo preentrenadas como Resnet18 y EfficientNet-B0.

Para la elección de las máquinas de aprendizaje profundo CNN a implementar se tuvieron en cuenta diferentes factores, primeramente, los antecedentes consultados [5][9][11] que eligieron esta arquitectura para sus máquinas de aprendizaje profundo

alcanzando desempeños superiores al 80%, teniendo en cuenta que estas fueron diseñadas por sus autores y no emplearon arquitecturas previamente concebidas. Adicionalmente, se puede aprovechar el uso de *Transfer Learning* de máquinas previamente entrenadas con una base de datos más amplia y generalizada que la implementada en este proyecto.

3.4.2.1 Espectrogramas

Dado que en este proyecto también se implementarán máquinas de aprendizaje profundo para la clasificación de PD, por lo cual se optó por emplear Mel-espectrogramas por medio de ventanas Hanning con 2048 muestras y un desplazamiento de 512 muestras, al audio completo, ya que estos representan de manera bi-dimensional la información espectro-temporal de las señales de audio, además son ampliamente utilizados en arquitecturas de CNN. Cabe resaltar que los espectrogramas se extrajeron de los audios debidamente preprocesados al igual que en el proceso que se realizó para la extracción de características.

Múltiples antecedentes anteriormente mencionados como [5] [47], entre otros, hacen uso de esta representación de entrada para las máquinas de aprendizaje profundo, obteniendo grandes resultados en tareas de clasificación de PD y pacientes sanos. Por estas razones se emplearon los espectrogramas como representación de entrada, ya que se considera por sus antecedentes en este tipo de aplicaciones una alternativa adecuada y efectiva para tareas relacionadas con clasificación de enfermedades patológicas como lo es el PD.

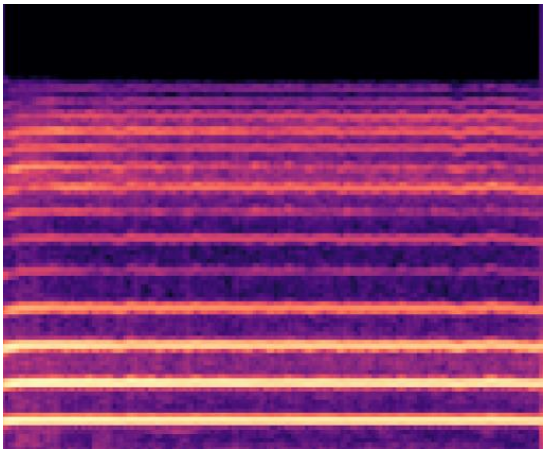

Figura 3-6 Espectrograma extraído para el proceso de clasificación. Fuente: Propia.

3.4.2.2 CNN ResNet 18 (Residual Network)

Se implementa una red neuronal convolucional (CNN), haciendo uso de una arquitectura previamente entrenada como ResNet (Figura 3 - 7). Para la implementación de esta máquina se debe hacer uso de *Transfer Learning*, que es aprovechar los pesos preentrenados de una base de datos grande como *ImageNet,* compuesta por un millón de muestras y entrenada para mil clases. Se tuvo que adaptar esta máquina a este proyecto debido a limitación de muestras en la base de datos, los espectrogramas extraídos en tres dimensiones RGB se remplazan como la entrada de la red, además de la capa de salida Fully Connected, imponiendo clasificación binaria [30].

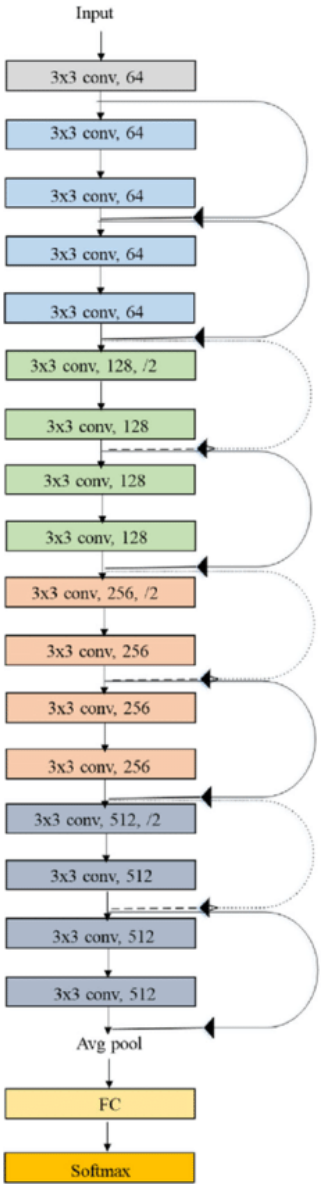

Figura 3-7 Arquitectura ResNet 18. Fuente: [30]

Esta ResNet 18, la cual se ilustra la Figura 3 - 7 está compuesta por 18 capas, donde 17 de ellas son convolucionales, seguida de una capa completamente conectada y una softmax para la clasificación final. Las capas convolucionales utilizan filtros 3 x 3. Cuando el tamaño del mapa de características de salida cambia, el tamaño de los filtros lo hace inversamente proporcional, aunque si el mapa se mantiene, los filtros también lo hacen. El submuestreo de esta máquina emplea capas convolucionales paso de dos, que permite eficiencia computacional e integridad en el flujo de características. Posteriormente se aplica una operación de pooling global, antes de pasar a las últimas capas [30].

3.4.2.3 CNN EfficientNet-B0

La arquitectura está compuesta por bloques MVConv (Mobile Inverted Bottleneck Convolution), donde se hace uso del mecanismo Squeeze-and-Excitation (SE), permitiendo así una extracción de características con menor costo computacional. La composición de esta máquina inicia con un bloque convolucional 3x3, haciendo uso de un Batch Normalization (BN), para estabilizar y acelerar el entrenamiento, seguido de varios bloques MBConv desde las primeras capas hasta las capas más profundas con diferentes kernels y expansiones como la capa de Global Average pooling, la cual se ayuda a reducir el número de parámetros, y finalizando con una capa densa totalmente conectada [32][48] .

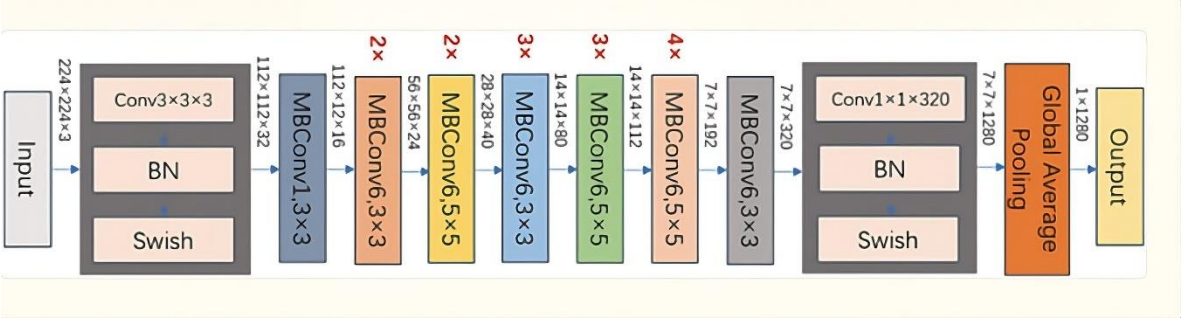

Figura 3-8 Arquitectura EfficientNet-B0. Fuente: [32]

Para funciones de activación, esta máquina emplea en su mayoría a *Swish*, que es una función no lineal, además de *Sigmoid*, que es una función de activación que ha demostrado grandes resultados en clasificación de imágenes. Por otro lado, la base de datos empleada para entrenamiento de esta arquitectura fue *ImageNet*, que contiene 1.2 millones de imágenes clasificadas para mil categorías[48].

3.5 Discusión

Las etapas implementadas en este proyecto tienen como objetivo implementar máquinas de aprendizaje superficial con cuatro máquinas como RF, DT, KNN y SVM, también para aprendizaje profundo con dos máquinas como ResNet18 y EfficientNet-B0. Para ambos casos, se realizó preprocesamiento de audio como filtrado, eliminación de silencios, recorte de audio a una misma longitud, eliminación de offset y remuestreo a 16k Hz. Estas subetapas contribuyen a disminuir el ruido, mantener homogeneidad en los audios a clasificar, y preservar las características

relevantes dentro de los mismos. En las máquinas de aprendizaje superficial, resulta un poco más relevante debido a la necesidad de la extracción de características numéricas, mientras que para el aprendizaje profundo se podría aceptar cierto ruido, sin embargo, el preprocesamiento de audio que puede afectar positivamente los resultados de clasificación, dando como resultado un espectrograma menos ruidoso y complejo de clasificar.

Para el caso de aprendizaje superficial, la segmentación por medio de ventanas para la extracción de características resulta conveniente, puesto que la base de datos contiene únicamente 470 audios y a partir de este método se aumenta la cantidad de muestras, que favorece positivamente a la generalización de los modelos y reduce los riesgos en sobreajuste. Por otro lado, debido a la robustez de las máquinas de aprendizaje profundo implementadas por *Transfer Learning*, la representación de los audios como datos de entrada fueron Mel-espectrogramas extraídos de los audios completos, que permiten identificar patrones para clasificación.

4. Pruebas y Resultados

4.1 Introducción

En esta sección se presentan los resultados obtenidos luego del desarrollo del proyecto (Capítulo 3) para la detección de Parkinson en grabaciones de audio; preprocesamiento de audio, la extracción de las características relevantes para la discriminación de esta enfermedad y la implementación de las máquinas de aprendizaje superficial como K-Nearest Neighbors (KNN), Suport Vector Machine (SVM), Decision Tree (DT) y Random Forest (RF) y profundo con Resnet-18 y EfficientNet-B0. Por otro lado, en este apartado se presentan los desempeños obtenidos. Asimismo, se realiza el análisis de resultados mediante diferentes métricas como la matriz de confusión, el reporte de clasificación con exactitud, sensibilidad, F1-Score y precisión.

4.2 Selección de características

Como se precisó anteriormente en el apartado 3.3.3, la selección de características para las máquinas de aprendizaje superficial fue mediante el método de *Wrappers* a través de *Decision Tree,* sin embargo, se tomaron en cuenta únicamente 1 millón de iteraciones, ya que solo el costo computacional de esta ascendía las 12 horas de procesamiento, y realizarlo para 21 millones podría resultar inviable. A pesar de que este procedimiento no garantiza la mejor combinación de características posible, con la cantidad de combinaciones aleatorias se puede acercar a la ideal. Se obtuvo como resultado las diez mejores características de las 51 extraídas:

Tabla 4-1 Características seleccionadas mediante el método Wrappers

Características seleccionadas
Frecuencia fundamental
Asimetría
Contraste espectral
Amplitud promedio
MFCC_7
MFCC_8
DeltaMFCC_11
DeltaDeltaMFCC_1
Shimmer
HNR

4.3 Resultados máquinas aprendizaje superficial

Para realizar la evaluación de los resultados obtenidos por los clasificadores de aprendizaje superficial, se particionaron los datos de entrenamiento y de prueba, mediante la proporción de 70% para entrenamiento y 30%, mediante la función *train_test_split,* Sin embargo, no se implementó estratificación en la partición de los datos. Además, se implementa la opción de rechazo, con el fin de aumentar la confiabilidad en las predicciones realizadas con un umbral de rechazo de 0.75. Por otro lado, para la clasificación final de pacientes se realiza una votación de las ventanas previamente clasificadas y la clase que mayor votación obtenga, se clasifica por la misma, otro aspecto para tener en cuenta es que, si existe una igualdad de votación, el dato se rechaza.

4.3.1 Resultados K-Nearest Neighbors (KNN)

En la implementación de la máquina KNN independiente del paciente, se tuvo en cuenta el método *Wrappers* para selección de características, además de los

posibles hiperparámetros mediante *GridSearch* expuestos en el apartado 3.4.1.3 con 5 K-folds, permitiendo un equilibrio entre el costo computacional y el rendimiento expuestos. Luego de este proceso, las métricas seleccionadas fueron: Número de características: 10, métrica de KNN: euclidiana, el número de vecinos: 9, y pesos: uniformes (ver Tabla 4-2).

Tabla 4-2 Hiperparámetros seleccionados para KNN mediante *GridSearch.*

Hiperparámetros KNN	
Número de características	10
Métrica de KNN	Euclideana
Número de vecinos	9
Peso de vecinos	Uniforme

Los resultados obtenidos por el modelo KNN, se presentan en la Tabla 4 - 3 y en la Figura 4 - 1. Al revisar estas métricas, se observa que los pacientes que padecen PD presentan una precisión aproximadamente 2% inferior en comparación con pacientes HC. Sin embargo, en cuanto a sensibilidad que identifica instancias positivas podemos visualizar lo contrario PD es 2% superior de los pacientes HC. Por otro lado, se obtuvo un valor de *F1-Score* bastante similar para ambas clases, eso debido a la compensación entre sensibilidad y precisión. Por último, en cuando a exactitud general del modelo en todas las instancias se obtuvo un 92.45%.

Tabla 4-3 Reporte de clasificación obtenida para KKN por ventanas.

	Precisión	**Sensibilidad**	**F1-Score**
HC	93.48%	91.27%	92.36%
PD	91.47%	93.63%	92.54%
Promedio	92.47%	92.45%	92.45%
Exactitud	92.45%		

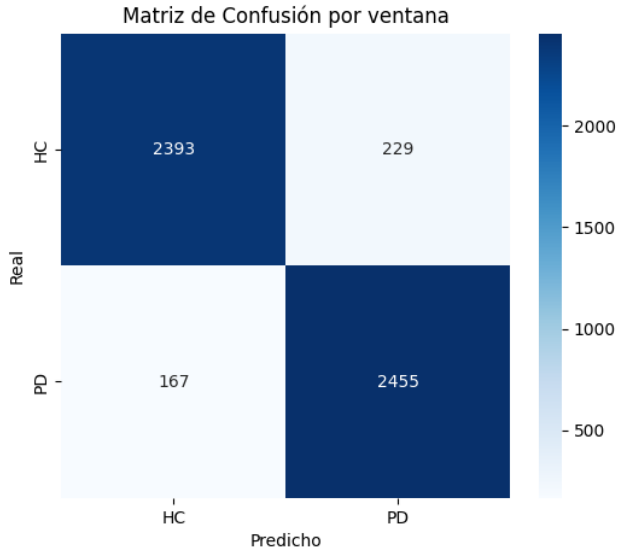

Figura 4-1 Matriz de confusión obtenida del modelo de KNN por ventanas.

Como se puede observar para este clasificador, los resultados obtenidos de las métricas del reporte de clasificación fueron consistentes. En exactitud general se obtuvo 96.38%, lo cual hace efectiva la clasificación mediante KNN en clasificación de pacientes. Al comparar el rendimiento obtenido por ventanas se puede observar una mejora de 2% en cuanto al rendimiento en general. Por otra parte, al ser una aplicación médica la métrica que mayor apreciación requiere es la sensibilidad, con el fin de evitar Falsos Negativos, así como se observa en la Figura – 42, los Falsos Negativos fueron 3, y 2 Falsos Positivos. Otro aspecto para tener en cuenta es la opción de rechazo que obtuvo como resultado 473 de 5244 por ventanas, siento esto el 8.8% de las ventanas rechazadas, luego de votación mediante ventanas por audio, no se rechaza ningún sujeto. También cabe resaltar el resultado obtenido de exactitud con datos de entrenamiento con 98.88%.

Tabla 4-4 Reporte de clasificación obtenida para KKN independiente del paciente.

	Precisión	**Sensibilidad**	**F1-Score**
HC	95.71%	97.10%	96.40%
PD	97.06%	95.65%	96.35%
Promedio	96.38%	96.38%	96.38%
Exactitud	96.38%		

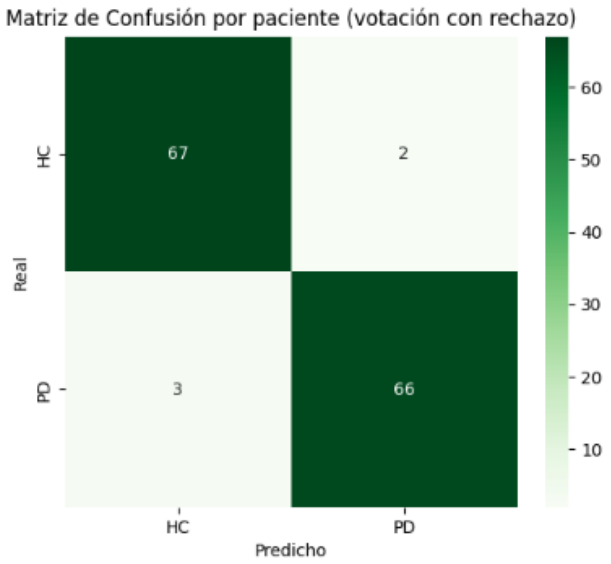

Figura 4-2 Matriz de confusión obtenida del modelo de KNN independiente del paciente.

4.3.2 Resultados Support Vector Machine (SVM)

Las máquinas SVM implementadas independientemente del paciente, se empleó el método Wrappers para selección de características, junto con los posibles hiperparámetros expuestos en el apartado 3.4.1.2 mediante *GridSearch* con 5 K-folds, Tras este proceso, los seleccionados de mejores métricas para datos separados fueron: Número de características: 10, el parámetro *C* fueron: 0.1, gamma: Scale, y kernel: RBF como se aprecia en la Tabla 4 - 5.

Tabla 4-5 Hiperparámetros seleccionados para SVM mediante *GridSearch*.

Hiperparámetros SVM	
Número de características	10
C	0.1
Gamma	Scale
Kernel	RBF

Las métricas obtenidas mediante ventanas reportadas en la Tabla 4-6, así como en la Figura 4 -3, se puede observar que se obtuvieron mejores resultados de pacientes con PD con respecto a pacientes HC, puesto que los Falsos Positivos fueron altos, alcanzando una sensibilidad del 90.96%, representando un aproximado del 10% de las ventanas clasificadas. Por otra parte, cabe destacar lo obtenido en los Falsos Negativos, a pesar de ser una métrica bastante sensible y de gran importancia únicamente se obtuvieron 141 ventanas mal clasificadas, con un 95.62% de sensibilidad. También cabe mencionar, que se obtuvieron 445 de 5244 ventanas

rechazadas basándonos en el umbral de rechazo del 0.75, así como la exactitud de los datos de entrenamiento fue 94.55%

Tabla 4-6 Reporte de clasificación obtenida para SVM por ventanas.

	Precisión	Sensibilidad	F1-Score
HC	94.42%	90.96%	92.66%
PD	91.28%	94.62%	92.92%
Promedio	92.85%	92.79%	92.79%
Exactitud	92.79%		

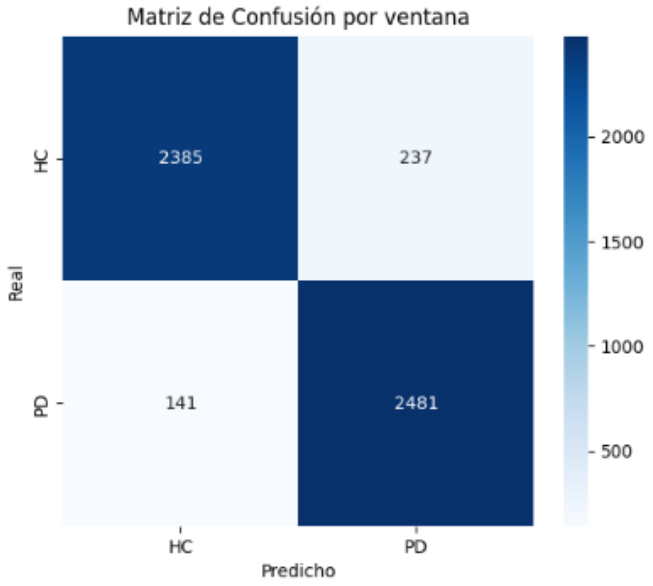

Figura 4-3 Matriz de confusión obtenida del modelo de SVM por ventanas.

En la Tabla 4 – 7 se presentan los resultados obtenidos por audio de cada paciente donde se puede observar que las métricas presentan una mejora luego de haber realizado la votación por ventanas en general, precisamente se puede visualizar que las métricas de sensibilidad en pacientes HC aumenta un 5% aproximadamente, así como la exactitud general de la máquina. Con respecto a los Falsos Positivos como Falsos Negativos, se obtuvo una clasificación errónea de 3 audios para cada uno.

Tabla 4-7 Reporte de clasificación obtenida para SVM independientemente del paciente

	Precisión	Sensibilidad	F1-Score
HC	95.65%	95.65%	95.65%
PD	95.65%	95.65%	95.65%
Promedio	95.65%	95.65%	95.65%
Exactitud	95.65%		

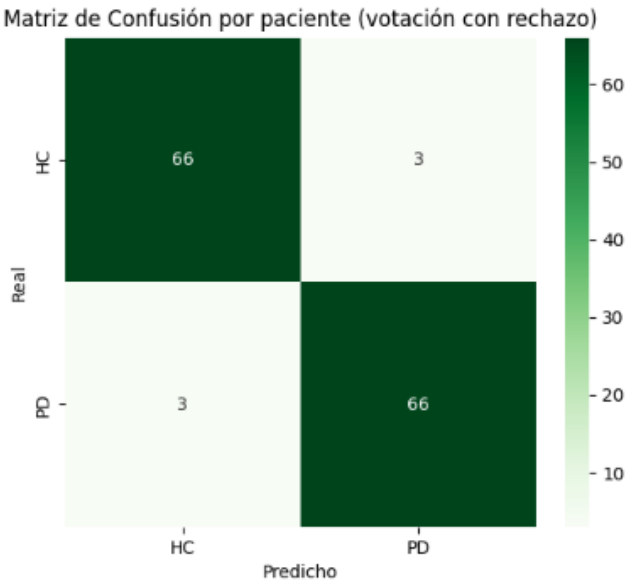

Figura 4-4 Matriz de confusión obtenida del modelo de SVM independientemente del paciente.

4.3.3 Resultados Decision Tree (DT)

Para las dos máquinas DT implementadas independientemente del paciente, se utilizó para selección de características el método Wrappers, junto los posibles hiperparámetros expuestos en el apartado 3.4.1.1 mediante *GridSearch,* con 5 K-folds. A raíz de este proceso las métricas seleccionadas para datos separados fueron: Número de características: 10, criterio de decisión: Entropy, máxima profundidad de árbol: 20, mínimo de muestras por hoja: 1, y mínimo de muestras para dividir: 2.

Tabla 4-8 Hiperparámetros seleccionados para DT mediante *GridSearch*.

Hiperparámetros DT	
Número de características	10
Criterio de decisión	Entropy
Máxima profundidad del árbol	20
Mínimo de muestras por hoja	1
Mínimo de muestras para dividir	2

Los resultados obtenidos mediante este clasificador DT por las ventanas de tiempo están plasmados en la siguiente tabla, donde en términos generales se obtuvo un buen rendimiento alcanzando un 89%. Como se puede observar se presenta una sensibilidad similar en pacientes HC con respecto a pacientes PD, que es donde se busca la baja tasa de pacientes clasificados incorrectamente.

Tabla 4-9 Reporte de clasificación obtenida para DT por ventanas.

	Precisión	**Sensibilidad**	**F1-Score**
HC	88.92%	89.09%	89.01%
PD	89.07%	88.90%	88.99%
Promedio	89.00%	89.00%	89.00%
Exactitud	89.00%		

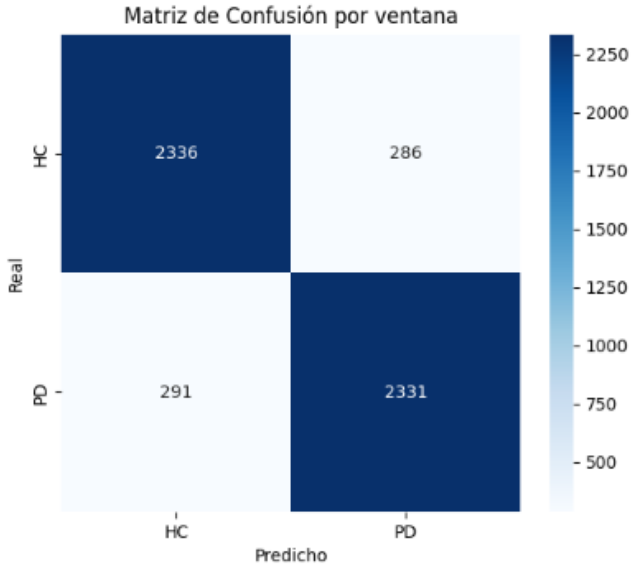

Figura 4-5 Matriz de confusión obtenida del modelo de KNN por ventanas

Los resultados obtenidos luego de realizar la votación mediante las ventanas extraídas presentan una mejora considerable en el rendimiento del clasificador, puesto que luego de realizar la votación se observa una mejora de 5% aproximadamente en las métricas obtenidas. También cabe resaltar la confiabilidad de los datos donde se obtuvieron 411 de 5244 ventanas que fueron rechazados por el umbral de opción de rechazo implementado siendo el 7.8%, sin embargo, no hubo pacientes que se rechazaran luego de haber hecho la votación por ventanas. Por otro lado, la exactitud general del sistema con datos de entrenamiento fue de 95.89%

Tabla 4-10 Reporte de clasificación obtenida para DT independiente del paciente.

	Precisión	Sensibilidad	F1-Score
HC	94.37%	97.10%	95.71%
PD	97.01%	94.20%	95.59%
Promedio	95.65%	95.65%	95.65%
Exactitud	95.65%		

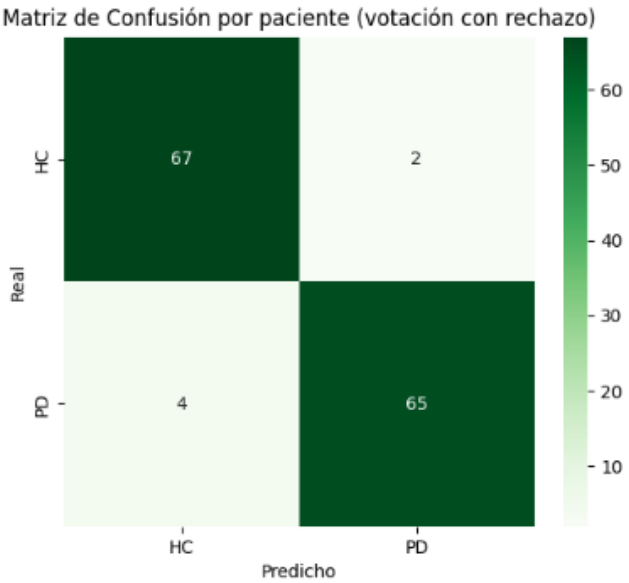

Figura 4-6 Matriz de confusión obtenida del modelo de DT independiente del paciente.

4.3.4 Resultados Random Forest (RF)

Para obtener los resultados de esta máquina de aprendizaje se tomó en cuenta el método de *Feature Importance*, con el fin de discriminar las características más determinantes para clasificar pacientes con PD y pacientes HC; para este caso en

específico se tomaron 10 características con los mejores resultados, las características seleccionadas fueron: Asimetría, ZCR, MFCC_9, Shimmer y MFCC_8, amplitud máxima, frecuencia fundamental, MFCC_10, MFCC_6, MFCC_7. Por otro lado, se tomó en cuenta *GridSearch* con 5 K-folds con el fin de discriminar los hiperparámetros expuestos en el apartado 3.4.1.4, los seleccionados con mejores rendimientos fueron: Bootstrap en False, máxima profundidad del árbol 20, mínimo de muestras por hoja 1, mínimo de muestras para dividir 2 y número de estimadores 200 para datos aleatorios.

Tabla 4-11 Hiperparámetros elegidos por *GridSearch*.

Hiperparámetros RF	
	Datos separados
Número de características	10
Bootstrap	False
Máxima profundidad del árbol	20
Mínimo de muestras por hoja	1
Mínimo de muestras para dividir	2
Número de estimadores	200

Los resultados obtenidos por medio del clasificador RF por ventanas en la Tabla 4 - 13 y la Figura 4 - 6 se puede observar que la clasificación presenta un patrón similar a los otros algoritmos implementados, presentando un mejor rendimiento en los falsos positivos con respecto a los falsos negativos, sin embargo, aún la métrica de sensibilidad es bastante considerable con un promedio de 92.14%, así como la exactitud general del modelo con 92.14%

Tabla 4-12 Reporte de clasificación obtenido para RF por ventanas.

	Precisión	**Sensibilidad**	**F1-Score**
HC	93.47%	90.62%	92.02%
PD	90.90%	93.67%	92.26%
Promedio	92.18%	92.14%	92.14%
Exactitud	92.14%		

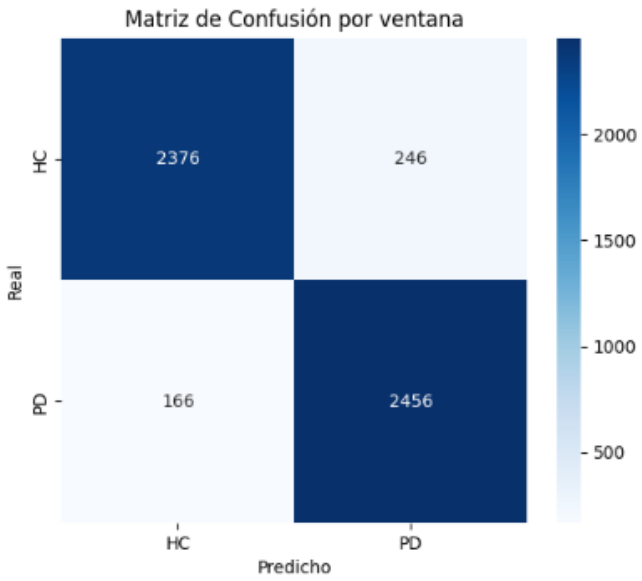

Figura 4-7 Matriz de confusión obtenida del modelo de RF por ventanas.

Realizando la votación de los audios mediante las ventanas se puede observar un patrón similar con respecto a los algoritmos pasados mejorando las métricas en general, aumentando la sensibilidad de los pacientes HC alrededor de un 5%, pues únicamente estarían 3 audios mal clasificados como Falsos Positivos, así como los Falsos Negativos con solo 4 audios únicamente. Por otro lado, la confiabilidad de los datos resultó afectada con 1024 datos rechazados de 5244 siendo el 19.5% el más alto comparado con los otros modelos implementados. Por otro lado, la exactitud general del sistema implementado fue de 98.45%.

Tabla 4-13 Reporte de clasificación obtenida para independiente del paciente.

	Precisión	**Sensibilidad**	**F1-Score**
HC	94.29%	95.65%	94.96%
PD	95.59%	94.20%	94.89%
Promedio	94.94%	94.93%	94.93%
Exactitud	94.93%		

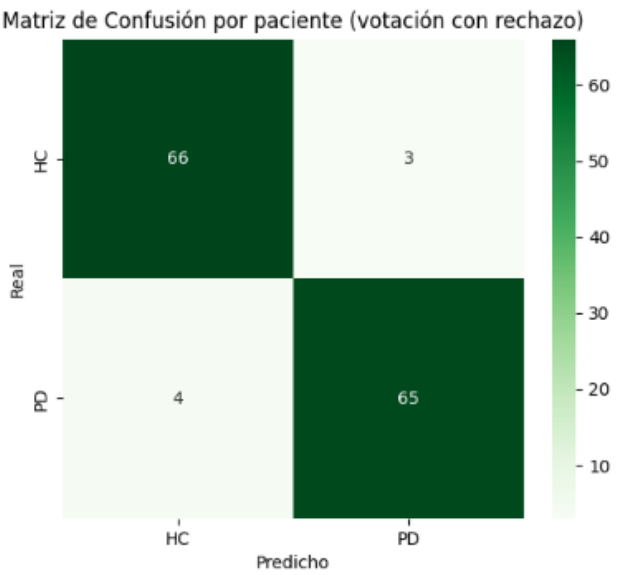

Figura 4-8 Matriz de confusión obtenida del modelo de RF independiente del paciente.

4.4 Resultados máquinas aprendizaje profundo

Para el caso de las máquinas de aprendizaje profundo ResNet-18 y EfficientNet-B0, las imágenes utilizadas como entrada fueron espectrogramas que se generaron a partir de los audios de las bases de datos. Para ser utilizados, estos fueron redimensionados a 224x224x3 en píxeles, con el fin de que fueran compatibles con las dos arquitecturas prestablecidas. Posteriormente, convertidas en tensores y pasando por el proceso de normalización [-1,1].

El conjunto de datos se particionó en un 70% para entrenamiento y 30% para prueba. Además, se aplicaron técnicas de *Data Augmentation*, también con el objetivo de mitigar el sobre entrenamiento en las máquinas implementadas, dentro de ellas se aplicaron: Rotación aleatoria de hasta 15°, rotación horizontal y variaciones de brillo y contraste mediante *ColorJitter* en Python.

Para estas máquinas en especial, se utilizaron los espectrogramas de todos los audios completos, alcanzando valores de métricas calculadas como precisión, sensibilidad, F1-Score y exactitud tuvieron desempeños que están por encima del 90%, e intentar alcanzar un rendimiento del 100% puede hacer perder a las maquinas la generalización si se prueban con otros datos, así como caer en sobreajuste de la base de datos de entrenamiento, debido a estos resultados no se utilizó la clasificación de ventanas extraídas de los audios.

4.4.1 Resultados CNN ResNet 18

Para los resultados de este proyecto se tomó en cuenta la máquina preentrenada en su totalidad, únicamente realizando cambios en los datos de entrada y la capa de salida que en este caso sería binaria. Primeramente, se pretendía emplear una arquitectura más profunda de este mismo tipo llamada ResNet 50, sin embargo, esta presentaba sobreajuste, que podría ser debido a diferentes causas como cantidad de audios o la baja complejidad de la base de datos seleccionada. Con el fin de evitar el congelamiento de muchas capas convolucionales, se optó por emplear la ResNet 18, siendo una arquitectura más ligera, pero robusta para nuestro caso en particular.

Tabla 4-14 Resultados reporte de clasificación CNN ResNet 18

	Precisión	**Sensibilidad**	**F1-Score**
HC	96.00%	96.00%	96.00%
PD	94.83%	94.83%	94.83%
Promedio	95.41%	95.41%	95.41%
Exactitud	95.49%		

Los resultados obtenidos por parte de esta arquitectura que se pueden apreciar en la Tabla 4 - 14, con PD en precisión de 94.83% inferior al 96.00% de pacientes HC, mientras que para sensibilidad se obtiene un 96.00% para pacientes HC y para PD un 94.83%. Obteniendo finalmente una exactitud de 95.49%. Revisando los resultados obtenidos en la Figura 4 – 9, se puede evidenciar que el rendimiento de la máquina con respecto Falsos Negativos o Flasos Positivos, cada uno obtuvo 3 audios mal clasificados. Por otro lado, la exactitud en entrenamiento fue del 97.89%. También cabe resaltar que luego de implementar la opción de rechazo, 8 audios fueron rechazados debido a la confiabilidad de los datos.

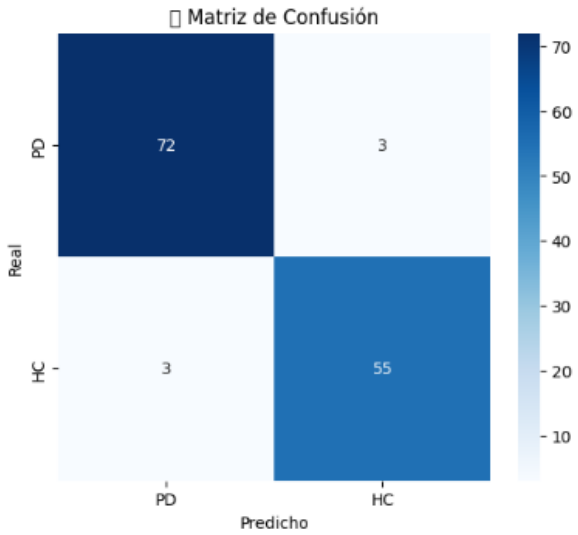

Figura 4-9 Matriz de confusión para CNN ResNet 18.

4.4.2 Resultados CNN EficientNetB0

Para esta arquitectura en particular, debido a que con la máquina completa se sobreentrenada el modelo, se optó por congelar las primeras dos MBConv, ya que las primeras capas son las encargadas de extraer características más superficiales. Además de esto, para la capa número 6 se realizó un *dropout* con el fin de prevenir también el sobreentrenamiento, lo cual desactiva aleatoriamente neuronas, lo que impide que se aprendan ciertos caminos o características.

Tabla 4-15 Resultados reporte de clasificación CNN ResNet 18

	Precisión	**Sensibilidad**	**F1-Score**
HC	98.44%	98.44%	98.44%
PD	98.67%	96.67%	98.67%
Promedio	98.55%	98.55%	98.55%
Exactitud	98.56%		

Por parte de EficientNet-B0 los resultados obtenidos son bastante similares entre clases, puesto que para HC se obtuvo una precisión de 98.41% y para PD 97.44%, presentando un caso similar tanto en sensibilidad, que es la metrica a tener más en cuenta por los Falsos Positivos y Negativos. Por consiguiente, se obtiene una exactitud general en el modelo de 97.87%, únicamente teniendo dos desaciertos sumando los Falsos Negativos y Falsos

positivos como se observa en la Figura 4 – 10. Por otro lado, se obtuvieron 2 muestras rechazadas de 141 representando el 1.4%.

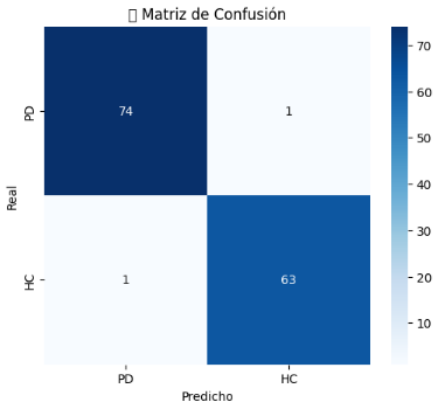

Figura 4-10 Matriz de confusión para CNN EfficientNet-B0.

4.5 Discusión de resultados

4.5.1 Discusión de resultados máquinas de aprendizaje superficial

Con el fin de realizar la evaluación y discusión de resultados obtenidos de las máquinas de aprendizaje superficial implementadas en este proyecto, se toman en cuenta los desempeños obtenidos dados en las Tablas 4 - 16 y 4 - 17 del apartado anterior de resultados. Adicionalmente, se pretende comparar estos rendimientos con los resultados obtenidos en los antecedentes previamente consultados.

Tabla 4-16 Tabla resumen resultados obtenidos con las máquinas de aprendizaje superficial.

Modelo	Precisión	Sensibilidad	F1-Score	Exactitud
KNN	96.39%	96.38%	96.38%	96.38%
SVM	95.65%	95.65%	95.65%	95.65%
DT	95.69%	95.65%	95.65%	95.65%
RF	94.94%	94.93%	94.93%	94.93%

Como se muestra en la tabla anterior, los resultados obtenidos en clasificación por cada una de las máquinas de aprendizaje superficial implementadas, donde se observa que los modelos KNN tuvo mejores rendimientos de manera general con respecto a SVM, DT y RF, a pesar de ser un rendimiento muy similar. Con respecto a la sensibilidad, se presentaron resultados bastante similares con un valor mayor

a 94% de rendimiento, lo que minimiza los errores de clasificación como Falsos Negativos o Falsos Positivos.

Los resultados obtenidos por KNN y RF presentan una pequeña diferencia en cuanto a los errores cometidos, puesto que KNN únicamente obtuvo 3 Falsos Negativos (Figura 4 - 2), mientras que RF obtuvo 4 (Figura 4 - 8). Por otro lado, para KNN en Falsos Positivos se obtuvieron 2 (Figura 4 - 2), mientras que RF obtuvo 3 (Figura 4 - 8). Por esto si se visualiza desde una perspectiva médica, se podría afirmar que KNN obtuvo un mejor rendimiento, ya que únicamente se identificaron 3 Falsos Negativos. Además, KNN únicamente rechazó 8.8% de las ventanas clasificadas, mientras que RF obtuvo un 19.5%, que le daría más confiabilidad a KNN como modelo de clasificación.

Otro aspecto para tener en cuenta son las similitudes entre los errores cometidos, luego de observar que los audios fueron clasificados erróneamente un Falso Positivo que resultó en todos los clasificadores fue FC\VI1, los demás Falsos Positivos variaron entre los algoritmos. Por otro lado, para los Falsos Negativos se obtuvo a AC VO1, VA1, VA2 para todos los clasificadores, exceptuando a RF, en los demás audios se obtuvieron resultados diversos entre sí.

Tabla 4-17 Tabla resumen mejores resultados antecedentes con máquinas de aprendizaje superficial.

Antecedente	Máquina	Precisión	Sensibilidad	F1-Score	Exactitud
[7]	**FNN**	99.11%	98.78%	99.96%	99.23%
[9]	**MLP**	100%	83%	91%	90%
[12]	**AdaBoost**	91%	87%	89%	84.5%
[13]	**RF**	98.90%	98.54%	98.63%	98.58%
[14]	**KNN**	83%	81%	80%	85%

Comparando estos resultados con el sistema desarrollado, se observa que el clasificador KNN logró un rendimiento del 96.38% en cuanto a sensibilidad, donde comparando con los mejores modelos consultados en [6] y [12], este sería inferior en un aproximado de 6%.

Por otro lado, los antecedentes analizados obtienen diversidad de modelos con las mejores métricas, cada uno obteniendo desempeños altos con diferentes algoritmos. Sin embargo, comparando el clasificador implementado con unos de los antecedentes [9], [12], [14], se puede evidenciar una mejora en cuanto a la exactitud general, no obstante, cabe resaltar que en cada proyecto realizado se tomaron en cuenta diferentes tipos de características o diferentes forma de extracción, así como

diferentes estrategias de emplear las máquinas, lo cual también puede afectar el rendimiento de las mismas.

4.5.2 Discusión de resultados máquinas de aprendizaje profundo

Con el fin de realizar la evaluación y discusión de resultados obtenidos de las máquinas de aprendizaje profundo implementadas en este proyecto, se toman en cuenta los desempeños obtenidos dados en las Tablas del apartado anterior de resultados. Adicionalmente, se pretende comparar estos rendimientos con los resultados obtenidos en los antecedentes previamente consultados.

Tabla 4-18 Tabla resumen resultados obtenidos con las máquinas de aprendizaje profundo.

Antecedente	Modelo	Precisión	Sensibilidad	F1-Score	Exactitud
Propia	**EficientNet-B0**	98.55%	98.55%	98.55%	98.56%
Propia	**Resnet 18**	95.41%	95.41%	95.41%	95.49%
[9]	**CNN**	75%	100%	86%	80%
[11]	**LSTM-CNN**	92%	92%	92%	91.7%
[5]	**CNN**	0.97 AUC			
[14]	**CNN**	82%	87%	75%	79%

Como se observa en la Tabla 4 – 18, los antecedentes previamente revisados presentan diferentes resultados en cuando a rendimiento en sus máquinas de aprendizaje profundo. En [11] destaca su resultado obtenido de exactitud con un 91.7%, además podemos observar un 0.97 en AUC, que se puede decir que presenta un buen rendimiento, a pesar de no haber implementado diferentes métricas de desempeño.

En cuanto a las máquinas implementadas en este proyecto, se observa que la máquina EfficientNet-B0 presenta un rendimiento más alto que ResNet18, a pesar de haber sido sometida al congelamiento de dos de sus capas y haber hecho el *Drop Out* en una de sus capas profundas, mientras que para ResNet18 se implementó la arquitectura en su totalidad, esto demuestra la solidez que puede llegar a lograr EfficientNet-B0 en cuanto a desempeño. Otro aspecto que se observa según las matrices de confusión de estas dos máquinas es la robustez en cuanto a los Falsos Negativos que obtuvo EfficientNet-B0 con únicamente 1 solo audio mal clasificado y este proyecto ante el hecho de ser una aplicación médica se debe minimizar los errores de clasificación, caso contrario con ResNet18 con 3 Falsos Negativos, un rendimiento menor con respecto a su similar. Además, la cantidad de audios rechazados por parte de ResNet18 con 8 representando un 5.67% de los

audios probados, mientras que EfficientNet únicamente 2 que sería el 1.41% de los datos.

Realizando la comparación de las máquinas implementadas con respecto a los antecedentes consultados, se puede apreciar un mejor resultado global con 98.56% en EfficientNet-B0 y con 95.49% en ResNet-18 de exactitud en las máquinas empleadas en este proyecto. Sin embargo, en los estudios anteriores, sus arquitecturas fueron diseñadas por sus autores desde cero, mientras que en este proyecto se implementó la técnica de Transfer Learning que resulta importante debido al costo computacional bajo, puesto que este viene con un preentrenamiento bastante completo por medio de ImageNet.

4.6 Alcances y limitaciones

Con base en el análisis de resultados del apartado 4 y de cada una de las pruebas realizadas al sistema compuesto de las etapas de preprocesamiento, extracción de características y clasificación, se establecen los principales alcances y limitaciones a continuación.

4.6.1 Alcances

- Los algoritmos desarrollados a lo largo de este proyecto se encuentran con la capacidad de clasificar pacientes con PD y pacientes sanos con nuevos datos en un ambiente académico y que cuente con datos de vocales en italiano, por lo que resulta la necesidad de seguir investigando y mejorando los rendimientos obtenidos, así como su implementación de diferentes idiomas y sonidos emitidos que es un factor que puede afectar su rendimiento.

- Las máquinas de aprendizaje implementadas, tanto superficiales como profundas presentaron un rendimiento superior al 85%, lo cual se puede clasificar con una efectividad superior a este valor pacientes con PD y pacientes sanos.

- La implementación de arquitecturas prestablecidas de redes neuronales convolucionales como ResNet-18 y EfficientNet-B0, adaptadas para esta aplicación en particular, alcanzaron resultados en clasificación sin grandes costos computacionales con rendimientos superiores a 95%.

4.6.2 Limitaciones

- El tamaño reducido de la base de datos, puesto que no se cuenta con audios en diferentes idiomas, que puedan proveer una mayor generalización para la clasificación de la presencia y ausencia de Parkinson en audios de voz.

- Los modelos pueden verse afectados por factores externos, puesto que grabaciones de audio que presenten ruido de fondo, micrófono defectuoso, entre otros, pueden llegar a ser factores cruciales y afectar negativamente los resultados.

- El sistema solamente se limita a la detección de la enfermedad y no logra identificar el grado de avances de la enfermedad de Parkinson en un ámbito netamente académico. Además de esto, el modelo no ha sido evaluado en un entorno clínico real ni comparado con diagnósticos médicos.

- El modelo no muestra necesariamente los mismos rendimientos para cada sujeto, como sucede en las máquinas de aprendizaje superficial obteniendo una clasificación diferente para los mismos audios, además puede haber diferencias en la voz, sexo, acento, entre otras que pueden influir en las predicciones.

- Los modelos aplicados no presentan la capacidad de recibir datos en tiempo real de pacientes, con el fin de tener una posible clasificación de pacientes de manera inmediata, lo que limita la accesibilidad de personas a este sistema.

5. Conclusiones Generales

Este trabajo de grado tuvo como objetivo el desarrollo de un sistema que clasificara pacientes con Parkinson y pacientes sanos mediante grabaciones de voz, empleando técnicas de procesamiento de señales y reconocimiento de patrones. Este objetivo fue llevado a cabo mediante el preprocesamiento de señales de audio, extracción de características, clasificación de pacientes, y análisis de resultados. Del sistema en general se concluye lo siguiente:

- Mediante el proyecto desarrollado se logró determinar la presencia o ausencia de la enfermedad de Parkinson en audios de voz, utilizando técnicas de procesamiento de señales digitales como eliminación de offset y silencios, normalización y filtrado. Este proceso fue respaldado por la implementación de máquinas de aprendizaje superficial como Decision Tree, Random Forest, K-Nearest Neigbohrs y Support Vector Machine y profundo como ResNet18 y EfficientNet-B0.
- Se determinó que las máquinas de aprendizaje superficial tienden a aumentar al menos un aproximado del 2% de manera general en métricas como precisión, sensibilidad, F1-Score y exactitud luego de realizar votación mediante ventanas de tiempo para cada audio de voz en la detección de presencia o ausencia de Parkinson.
- El proceso de TransferLearning en redes neuronales convolucionales (CNN) es una estrategia efectiva para la detección de presencia o ausencia de Parkinson en audios de voz, debido al previo entrenamiento con la base de datos *ImageNet* de gran volumen*,* pues es posible que dentro de esta hayan imágenes con rasgos o características similares a los espectrogramas, demostrado con el rendimiento logrado por el modelo EfficientNet-B0 de 98.55% de sensibilidad, logrando solo un Falso Negativo y un Falso Positivo, a pesar de las capas ocultadas y la aplicación de Dropout.
- Se desarrollaron varios sistemas de aprendizaje superficial para la detección de presencia o ausencia de Parkinson, entre ellos Decision Tree, Random Forest, K-Nearest Neigbohrs y Support Vector Machine, obteniendo como mejor resultado a KNN con una sensibilidad de 96.38% contando con dos Falsos Negativos en clasificación y con únicamente el 8.8% de ventanas rechazadas.
- Se determinó que el camino de selección de características por parte de Random Forest también resulta bastante discriminante comparado con el método Wrappers para la detección de presencia o ausencia de Parkinson

en audios de voz, puesto que los rendimientos obtenidos por parte de las máquinas de aprendizaje superficial implementadas como Decision Tree, Support Vector Machine, K-Nearest Neighbors y Random Forest fueron superiores al 94% en las métricas extraídas de precisión, sensibilidad, F1-Score y exactitud, obteniendo únicamente cuatro y dos Falsos Negativos respectivamente, además de la similitud de características discriminantes seleccionadas como: Asimetría, MFCC_7, MFCC_8, Shimmer y Frecuencia fundamental, que se podrían apreciar como las más relevantes de acuerdo a la concordancia en la selección por parte de ambos métodos.

- La integración de diferentes tipos de características como ceptrales, frecuenciales, temporales y calidad vocal resulta determinante para la detección de presencia o ausencia de Parkinson en audios de voz, puesto que las características seleccionadas como las más discriminantes pertenecían a los diferentes tipos extraídos, destacando que se obtuvieron cuatro ceptrales de diez seleccionadas.

6. Trabajos Futuros

Con base en el proyecto realizado se plantean las siguientes perspectivas para trabajos futuros:

- Primeramente se plantea una ampliación de la base de datos, con la incorporación de diferentes idiomas, donde los pacientes pronuncien vocales en varios lenguajes, permitiendo una visión más amplia para aplicaciones como la de este proyecto. Además, se propone tomar nuevas muestras a los pacientes a lo largo del tiempo con el fin de poder clasificar el grado de avance de la enfermedad.

- Explorar diferentes métodos de extracción de características como la implementación de técnicas avanzadas como EMD (Empirical mode descompostion), HSA (Hilber Spectral Analysis), entre otras, para evaluar el potencial en la discriminación de patrones de los datos.

- Realizar la extracción de espectrogramas por ventanas, con el fin de aumentar el número de datos a entrenar y clasificar, con el objetivo de desarrollar una arquitectura de una red neuronal por completo y que obtenga una mayor generalización para la detección de presencia y ausencia de Parkinson en audios de voz.

- Desarrollo de una aplicación móvil basada en máquinas de aprendizaje superficial, mediante TensorFlow Lite u ONNX Mobile, facilitando la implementación de estos algoritmos en dispositivos móviles, permitiendo una forma de detección que pueda proveer un primer filtro para el diagnóstico definitivo de la enfermedad de Parkinson que sirva como apoyo para el personal médico.

7. Referencias Bibliográficas

[1] Organización Mundial de la Salud (OMS), “Enfermedad de Parkinson,” *Nota descriptiva*, Aug. 09, 2023 Accessed: Sep. 20, 2025. [Online]. Available: https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease.

[2] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, “Suitability of Dysphonia Measurements for Telemonitoring of Parkinson’s Disease,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1015–1022, Apr. 2009, doi: 10.1109/TBME.2008.2005954.

[3] I. C. Lampropoulos, F. Malli, O. Sinani, K. I. Gourgoulianis, and G. Xiromerisiou, “Worldwide trends in mortality related to Parkinson’s disease in the period of 1994–2019: Analysis of vital registration data from the WHO Mortality Database,” *Front. Neurol.*, vol. 13, Oct. 2022, doi: 10.3389/fneur.2022.956440.

[4] S. Sharma and P. Singh, “Automated Detection of Parkinson’s Disease,” in *2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST)*, Feb. 2022, pp. 36–42. doi: 10.1109/ICRTCST54752.2022.9781972.

[5] A. Iyer *et al.*, “A machine learning method to process voice samples for identification of Parkinson’s disease,” *Sci. Reports* |, vol. 13, p. 20615, 123AD, doi: 10.1038/s41598-023-47568-w.

[6] Sonia Lizeth Alemán Pullas ; Carlos Xavier Montero Balarezo ; Eddy Xavier Díaz Recalde ; Christian Manuel Jarro Sanchez, “Enfermedad de Parkinson. Diagnóstico y tratamiento,” *RECIMUNDO*, vol. 1, pp. 250–266, Apr. 2022.

[7] S. Srinivasan, P. Ramadass, S. K. Mathivanan, K. Panneer Selvam, B. D. Shivahare, and M. A. Shah, “Detection of Parkinson disease using multiclass machine learning approach,” *Sci. Rep.*, vol. 14, no. 1, p. 13813, Jun. 2024, doi: 10.1038/s41598-024-64004-9.

[8] S. Sharanyaa, P. N. Renjith, and K. Ramesh, “Classification of parkinson’s disease using speech attributes with parametric and nonparametric machine learning techniques,” in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Dec. 2020, pp. 437–442. doi: 10.1109/ICISS49785.2020.9316078.

[9] I. K. Veetil, V. Sowmya, J. R. Orozco-Arroyave, and E. A. Gopalakrishnan, “Robust language independent voice data driven Parkinson’s disease detection,” *Eng. Appl. Artif. Intell.*, vol. 129, Mar. 2024, doi: 10.1016/j.engappai.2023.107494.

[10] S. Lih, O. Yuki, H. U. Raghavendra, and R. Yuvaraj, “A deep learning approach for Parkinson ’ s disease diagnosis from EEG signals,” *Neural Comput. Appl.*, vol. 5, 2018, doi: 10.1007/s00521-018-3689-5.

[11] M. Wodzinski, A. Skalski, D. Hemmerling, J. R. Orozco-Arroyave, and E. Noth, “Deep Learning Approach to Parkinson’s Disease Detection Using Voice Recordings and Convolutional Neural Network Dedicated to Image Classification,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2019, pp. 717–720. doi: 10.1109/EMBC.2019.8856972.

[12] M. A. Hossain and F. Amenta, “Machine Learning-Based Classification of Parkinson’s Disease Patients Using Speech Biomarkers,” *J. Parkinsons. Dis.*, vol. 14, no. 1, pp. 95–109, Jan. 2024, doi: 10.3233/JPD-230002.

[13] R. Lamba, T. Gulati, A. Jain, and P. Rani, “A Speech-Based Hybrid Decision Support System for Early Detection of Parkinson’s Disease,” *Arab. J. Sci. Eng.*, vol. 48, no. 2, pp. 2247–2260, Feb. 2023, doi: 10.1007/s13369-022-07249-8.

[14] G. Costantini *et al.*, “Artificial Intelligence-Based Voice Assessment of Patients with Parkinson’s Disease Off and On Treatment: Machine vs. Deep-Learning Comparison,” *Sensors*, vol. 23, no. 4, p. 2293, Feb. 2023, doi: 10.3390/s23042293.

[15] National Institute of Neurological Disorders and Stroke, “Parkinson’s Disease,” *U.S. Dep. Heal. Hum. Serv.*, Mar. 2025.

[16] GEN, “Parkinson’s Disease Brain Malfunction Evident in Scans Potentially Years Before Symptoms,” Jun. 19, 2019.

[17] K. Tjaden, “Speech and Swallowing in Parkinson’s Disease,” *Top. Geriatr. Rehabil.*, vol. 24, no. 2, pp. 115–126, Apr. 2008, doi: 10.1097/01.TGR.0000318899.87690.44.

[18] A. Antonini *et al.*, “Validation of the Italian version of the Movement Disorder Society—Unified Parkinson’s Disease Rating Scale,” *Neurol. Sci.*, vol. 34, no. 5, pp. 683–687, May 2013, doi: 10.1007/s10072-012-1112-z.

[19] Fernando L. Pagán, “Improving Outcomes Through Early Diagnosis of Parkinson’s Disease,” Sep. 2012. [Online]. Available: www.ajmc.com

[20] A. Rahman, S. S. Rizvi, A. Khan, A. A. Abbasi, S. U. Khan, and T. S. Chung, “Parkinson’s disease diagnosis in cepstral domain using MFCC and dimensionality reduction with SVM classifier,” *Mob. Inf. Syst.*, vol. 2021, 2021, doi: 10.1155/2021/8822069.

[21] B. Basant, N. Khodadadi, E. khodadadi, M. M. Eid, and S. K. Towfek,

"Advancements and Future Directions in Machine Learning for Medical Diagnostics: A Comprehensive Review," *J. Artif. Intell. Metaheuristics*, vol. 7, no. 2, pp. 18–31, 2024, doi: 10.54216/JAIM.070202.

[22] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[23] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *J. Data Anal. Inf. Process.*, vol. 08, no. 04, pp. 341–357, 2020, doi: 10.4236/jdaip.2020.84020.

[24] E. García-Gonzalo, Z. Fernández-Muñiz, P. García Nieto, A. Bernardo Sánchez, and M. Menéndez Fernández, "Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers," *Materials (Basel).*, vol. 9, no. 7, p. 531, Jun. 2016, doi: 10.3390/ma9070531.

[25] A. Khurana and O. P. Singh, "Estimation of Battery Life Cycle & Remaining using Machine Learning." Aug. 02, 2024. doi: 10.21203/rs.3.rs-4595067/v1.

[26] K. M. Alalayah, E. M. Senan, H. F. Atlam, I. A. Ahmed, and H. S. A. Shatnawi, "Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method," *Diagnostics*, vol. 13, no. 11, Jun. 2023, doi: 10.3390/diagnostics13111924.

[27] M. R. Camana Acosta, S. Ahmed, C. E. Garcia, and I. Koo, "Extremely Randomized Trees-Based Scheme for Stealthy Cyber-Attack Detection in Smart Grid Networks," *IEEE Access*, vol. 8, pp. 19921–19933, 2020, doi: 10.1109/ACCESS.2020.2968934.

[28] M. U. Hadi, R. Qureshi, A. Ahmed, and N. Iftikhar, "A lightweight CORONA-NET for COVID-19 detection in X-ray images," *Expert Syst. Appl.*, vol. 225, p. 120023, Sep. 2023, doi: 10.1016/j.eswa.2023.120023.

[29] A. Zaeemzadeh, N. Rahnavard, and M. Shah, "Norm-Preservation: Why Residual Networks Can Become Extremely Deep?," Apr. 2020, [Online]. Available: http://arxiv.org/abs/1805.07477

[30] F. Ramzan *et al.*, "A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks," *J. Med. Syst.*, vol. 44, no. 2, p. 37, Feb. 2020, doi: 10.1007/s10916-019-1475-2.

[31] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional

Neural Networks," Sep. 2020, [Online]. Available: http://arxiv.org/abs/1905.11946

[32] X. Chen *et al.*, "Application of EfficientNet-B0 and GRU-based deep learning on classifying the colposcopy diagnosis of precancerous cervical lesions," *Cancer Med.*, vol. 12, no. 7, pp. 8690–8699, Apr. 2023, doi: 10.1002/cam4.5581.

[33] D. T. M. S. Hagen Jaeger, "Mobile Device Voice Recordings at King's College London (MDVR-KCL) from both early and advanced Parkinson's disease patients and healthy controls," Sep. 2017.

[34] G. Dimauro, V. Di Nicola, V. Bevilacqua, D. Caivano, and F. Girardi, "Assessment of speech intelligibility in Parkinson's disease using a speech-to-text system," *IEEE Access*, vol. 5, pp. 22199–22208, Oct. 2017, doi: 10.1109/ACCESS.2017.2762475.

[35] T. Alkire and C. Rosen, *Romance Languages*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511845192.

[36] N. Madusanka and B. Lee, "Vocal Biomarkers for Parkinson's Disease Classification Using Audio Spectrogram Transformers," *J. Voice*, Dec. 2024, doi: 10.1016/j.jvoice.2024.11.008.

[37] Nilufar Niyozmatova, Kuanish Jalelov, Boymirzo Samijonov, and Mukhtaram Madrahimova, "Eliminating noise from a speech signal based on a pair of filters," *Int. J. Sci. Res. Arch.*, vol. 13, no. 2, pp. 401–410, Nov. 2024, doi: 10.30574/ijsra.2024.13.2.2058.

[38] C. García and D. Tapias, "LA FRECUENCIA FUNDAMENTAL DE LA VOZ Y SUS EFECTOS EN RECONOCIMIENTO DE HABLA CONTINUA," 2000 Accessed: Sep. 20, 2025. [Online]. Available: https://rua.ua.es/server/api/core/bitstreams/a631d1ee-5df5-4896-baee-da9ca6fd3e41/content.

[39] C. J. Chen, *Elements of human voice*. New Jersey: World Scientific, 2017. Accessed: Sep. 20, 2025. [Online]. Available: http://lccn.loc.gov/2016014277

[40] L. ESCOBAR, *Conceptos básicos de procesamiento digital de señales*, vol. 1. Ciudad de México: Universidad Nacional Autónoma de México, 2008. Accessed: Sep. 20, 2025. [Online]. Available: https://odin.fi-b.unam.mx/labdsp/files/libros/conceptos.pdf

[41] X. Peng, H. Xu, J. Liu, J. Wang, and C. He, "Voice disorder classification using convolutional neural network based on deep transfer learning," *Sci. Rep.*, vol. 13, no. 1, p. 7264, May 2023, doi: 10.1038/s41598-023-34461-9.

[42] M. KÖSEOĞLU and H. UYANIK, "Effect of Spectrogram Parameters and

Noise Types on The Performance of Spectro-temporal Peaks Based Audio Search Method," *Gazi Univ. J. Sci.*, vol. 36, no. 2, pp. 624–643, Jun. 2023, doi: 10.35378/gujs.1000594.

[43] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, "The choice of scaling technique matters for classification performance," Dec. 2022, doi: 10.1016/j.asoc.2022.109924.

[44] Y.-H. Wang *et al.*, "Identification of adaptor proteins using the ANOVA feature selection technique," *Methods*, vol. 208, pp. 42–47, Dec. 2022, doi: 10.1016/j.ymeth.2022.10.008.

[45] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2019.

[46] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," *Int. J. Data Sci. Anal.*, Feb. 2024, doi: 10.1007/s41060-024-00509-w.

[47] G. Costantini *et al.*, "Artificial Intelligence-Based Voice Assessment of Patients with Parkinson's Disease Off and On Treatment: Machine vs. Deep-Learning Comparison," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042293.

[48] S. Nigam, R. Jain, V. K. Singh, S. Marwaha, A. Arora, and S. Jain, "EfficientNet architecture and attention mechanism-based wheat disease identification model," *Procedia Comput. Sci.*, vol. 235, pp. 383–393, 2024, doi: 10.1016/j.procs.2024.04.038.