UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Análisis comparativo de métodos utilizados para distinguir entre voces humanas e imitaciones generadas por inteligencia artificial

Trabajo de graduación presentado por José Antonio Cayetano Molina González para optar a al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

GUATEMALA,

2024

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Análisis comparativo de métodos utilizados para distinguir entre voces humanas e imitaciones generadas por inteligencia artificial

Trabajo de graduación presentado por José Antonio Cayetano Molina González para optar a al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

GUATEMALA,

2024

Vo. Bo.

(f) _____

Ing. Luis R. Furlán

Tribunal Examinador:

(f) _____

Ing. Luis R. Furlán

(f) _____

(f) _____

Fecha de aprobación: Guatemala, _____ de _____ de 2024

Índice

Lista de cuadros vii
Lista de figuras ix
Resumen xi
Abstract xiii
I. Introducción 1
II. Justificación 3
III. Objetivos 5
A. General 5
B. Específicos 5
IV. Marco teórico 7
A. Inteligencia Artificial 7
B. Redes Neuronales 9
C. Filtros de audio 13
D. Audición humana 15
V. Metodología 19
A. Recolección de los datos para modelos los de *machine learning* 19
B. Preprocesamiento de los datos: 20
C. Creación de la encuesta: 21
D. Creación y entrenamiento de los modelos de *machine learning* 21
E. Ajustar hiperparámetros 23
VI. Resultados: 25
A. Análisis Exploratorio: 25
B. Modelo SVM 27
C. Otros modelos de machine learning 29
D. Modelo utilizando red neuronal LSTM 33
E. Reconocimiento humano 34

VII. Discusión 39
VIII. Conclusiones 45
IX. Bibliografía 47

Lista de cuadros

1. Cuadro 1 : Características de los conjuntos de datos disponibles en ASVspoof Challenge 2021 19
2. Cuadro 2 : Media de cada característica por cada etiqueta del conjunto de datos sin filtro aplicado 27
3. Cuadro 3 : Métricas de los diferentes *kernels* del modelo SVM para el conjunto de datos de prueba sin filtros. 27
4. Cuadro 4 : Métricas de los diferentes coeficientes “C” del modelo SVM con *kernel* radial, para el conjunto de datos de prueba sin filtros.....28
5. Cuadro 5 : Métricas de modelos base 30
6. Cuadro 6 : Métricas de los diferentes modelos de machine Learning creados con Autogluon 30
7. Cuadro 7 : Métricas del modelo LSTM con optimizador ADAM 33
8. Cuadro 8 : Métricas del examen de reconocimiento humano agrupado por edad .35
9. Cuadro 9 : Métricas del examen de reconocimiento humano agrupado por profesión 37

Lista de figuras

1. Figura 1 : Arquitectura de una neurona LSTM 11
2. Figura 2 : Arquitectura de la red neuronal LSTM 22
3. Figura 3 : Cantidad de datos por cada etiqueta (*deepfake* o bonafide) 25
4. Figura 4 : Distribución de la duración de los audios 25
5. Figura 5 : Distribución del *Bitrate* (Cantidad de información en el audio) 26
6. Figura 6 : Comparación de las métricas de los modelos SVM con diferente Kernel 28
7. Figura 7 : Comparación de métricas del modelo SVM con kernel radial y distintos valores de coeficiente C 29
8. Figura 8 : Comparación de los modelos de *machine learning* obtenidos con el Autogluon 31
9. Figura 9 : Importancia de las características para el conjunto de datos de prueba 31
10. Figura 10 : Comparación de los rendimientos promedio de los filtros estadísticos aplicados 32
11. Figura 11 : Comparación del modelo LSTM con diferentes pasos de aprendizaje 33
12. Figura 12 : Porcentaje de aciertos por cada pregunta del examen de reconocimiento humano 34
13. Figura 13 : Porcentaje de personas por cada edad 35
14. Figura 14 : Cantidad de personas por cada profesión 36
15. Figura 15 : Comparación de métricas de modelos con los datos del examen de reconocimiento humano 38

Resumen

Actualmente, la inteligencia artificial se ha implementado para diferentes usos, como la imitación de voces. Sin embargo, puede llegar un punto donde la imitación sea tan precisa que no se pueda distinguir entre una voz real y una generada por inteligencia artificial. Esto plantea implicaciones éticas y sociales, en donde la veracidad de una voz puede ser importante. Por eso, el propósito de esta investigación es comparar diferentes métodos para identificar voces generadas por inteligencia artificial y encontrar el más preciso y eficiente entre ellos.

Los métodos comparados incluyen modelos de *machine learning* utilizando filtros estadísticos, redes neuronales *Long Short-Term Memory* (LSTM), e incluso el reconocimiento humano. Para el entrenamiento de los modelos, se utilizó el conjunto de datos *Deepfake* proporcionado en la competencia de ASVspoof 2021, con la correspondiente extracción de características de audio. Utilizando audios de 2 segundos para todos los métodos, se encontró que las redes neuronales LSTM y algunos modelos de *machine learning* tuvieron el mejor desempeño, con una precisión por arriba del 97%, mientras que el reconocimiento humano tuvo un desempeño significativamente menor, con una precisión de aproximadamente 55%.

Estos resultados pueden proporcionar la base para la creación de herramientas de verificación de voz que mitiguen los daños causados por imitaciones. No obstante, esta investigación presenta ciertas limitaciones, como el uso de un conjunto de datos especifico y no diverso, además de utilizar audios de 2 segundos. Futuras investigaciones podrían explorar utilizar un conjunto de datos más diverso y duraciones de audio diferentes para poder evaluar que tan bien generalizan los modelos.

Abstract

The widespread implementation of artificial intelligence in various fields now includes voice imitation. However, the increasing precision of the deepfake voices could reach a point where distinguishing between real and synthetic voices could become a challenge. This raises ethical and social implications regarding instances where voice authentication is crucial. Therefore, the purpose of this study is to compare different methods to identify artificially generated voices and determine the most accurate and efficient among them.

The methods compared include machine learning models using statistical filters, Long Short-Term Memory (LSTM) neural networks, and human recognition. The models were trained using the Deepfake dataset provided by the ASVspoof 2021 challenge with the respective audio feature extraction. Using 2-second audio samples for all methods, the results show that LSTM networks and some machine learning models performed the best, with all metrics above 97%, while human recognition had a significantly lower performance with all metrics around 55%.

These results can provide a baseline for creating voice verification tools that mitigate the damage caused by voice imitation. However, this research comes with certain limitations, such as the use of a non-diverse dataset and the use of 2-second audio samples. Future research could explore using a more diverse dataset and different audio durations to better evaluate the performance of models.

I. Introducción

Las estafas causadas por imitación o personificación de voces son un problema creciente. Estas consisten en llamar a personas haciéndose pasar por seres queridos y pidiendo dinero o incluso información personal. La imitación de voz se logra mediante diferentes técnicas, siendo las redes neuronales una de las más utilizadas [1]. Este problema se ha intensificado debido a la viralización de herramientas de inteligencia artificial [2]. Esto plantea la necesidad de encontrar formas efectivas y eficientes de identificar voces generadas por inteligencia artificial y así prevenir posibles fraudes.

Desde la presentación de ChatGPT al público general, el uso de herramientas que basadas en inteligencia artificial ha crecido de manera exponencial. A esta innovación le tomó 5 días alcanzar un millón de usuarios, y para noviembre de 2023 se reportaron más de 180 millones de usuarios activos al mes. Incluso, en febrero del 2024 tuvo más de mil seiscientos millones de usuarios [3], casi el 20% de la población mundial. Esta popularización permitió que más personas tuvieran acceso a este tipo de herramientas basadas en inteligencia artificial.

Aunque la mayoría del tiempo estas herramientas son utilizadas para facilitar diversos aspectos de la vida, también presentan desafíos significativos. La facilidad de transmitir datos hoy en día contribuye a la difusión de nuevas aplicaciones de inteligencia artificial, sobre todo cuando estas se comparten en redes sociales y se vuelven una sensación.

Entre las herramientas que han ganado popularidad, la imitación de voces a través de redes neuronales destacó por las implicaciones que conlleva [4]. Esta tecnología se ha implementado para varias situaciones, desde lograr que una persona diga algo gracioso hasta la creación de una canción con la voz de un artista famoso. Sin embargo, en manos equivocadas, se ha convertido en una técnica para estafas. La forma en que logran esto es haciéndose pasar por un ser querido a través de una llamada telefónica, donde el atacante pide información personal o incluso dinero. Teniendo en cuenta que la voz humana es un atributo tan importante en la sociedad y se puede reconocer fácilmente, estas estafas son particularmente efectivas [5]. Además, reconocer la voz de una persona permite interpretar no solo las emociones que transmite, sino que además permite crear una imagen mental de la persona, lo cual genera más empatía hacia las peticiones del atacante [6].

Dado el impacto de estas estafas, resulta fundamental desarrollar métodos que permitan verificar si una voz fue generada por inteligencia artificial. Por ello el objetivo de esta investigación es comparar tres métodos para la identificación de voces generadas: modelos de *machine learning* con filtros estadísticos aplicados, redes neuronales *Long Short-Term Memory* (LSTM) y reconocimiento humano. Estos métodos se analizarán para determinar

cuál es el más preciso y eficiente, con el fin de establecer una base para la creación de sistemas que puedan prevenir las estafas mencionadas anteriormente, ya sea mediante intermediarios humanos o programas automatizados.

Para realizar la evaluación de los diferentes métodos se utilizó el conjunto de datos *Deepfake* proporcionado por ASVspoof 2021. Este conjunto incluye tanto audios con voces reales como audios con voces sintéticas o *deepfake.* Se utilizaron muestras de audio de 2 segundos para establecer un estándar que los modelo pudieran utilizar en su entrenamiento y evaluación. Para la creación de los modelos de *machine learning* y la red neuronal se utilizaron características extraídas de los audios y eso se utilizó para entrenar. Para lograr determinar cuál es el método más eficiente se utilizaron

El reconocimiento humano, por otro lado, se llevó a cabo mediante un examen que consistía en 10 preguntas dicotómicas. A los participantes se les presentaban audios y debían decidir si la voz era real o generada por inteligencia artificial. Este enfoque permite evaluar la capacidad de los humanos para distinguir entre voces reales y sintetizadas, proporcionando un punto de comparación respecto a los modelos creados. Además, se podrían analizar patrones que existan en la manera que los humanos contestan y de esta manera mejorar sistemas de verificación de voz.

Los resultados del estudio muestran que los humanos tienen una habilidad de reconocimiento para voces generadas por inteligencia artificial mucho más bajo que los modelos creados. Asimismo, varios modelos de *machine learning* y la red neuronal LSTM, obtuvieron resultados prometedores, indicando que son precisos para distinguir entre voces reales y sintéticas. Además, se evaluó la eficiencia de estos modelos en términos de consumo de recursos, y se encontró que algunos de los modelos de *machine learning,* sin ningún filtro estadístico aplicado mostraron mejores resultados.

II. Justificación

Durante los últimos años, las herramientas que utilizan Inteligencia Artificial para dar un servicio han aumentado considerablemente. Entre estas innovaciones, la imitación de voces con inteligencia artificial ha destacado por su capacidad de imitar las voces humanas con gran precisión. Sin embargo, esta tecnología ha demostrado ser una herramienta con potenciales beneficios y riesgos, ya que puede ser utilizada para aplicaciones creativas como para actividades fraudulentas. Esto resalta la necesidad de desarrollar métodos para identificar si una voz fue generada por inteligencia artificial o si es una voz real.

La imitación de voces no solo se emplea para poder recrear diálogos inexistentes, sino que también puede utilizarse para crear canciones con la voz de un artista famoso sin su consentimiento, lo cual constituye en una clara violación de derechos de autor. Un caso ampliamente conocido fue el de Bad Bunny, en donde un DJ utilizó su voz para crear una canción que se viralizo [7]. Además, los fraudes por clonación de voz se han hecho cada vez más comunes. Un estudio realizado por McAfee reveló que, de 7000 personas encuestadas, solo un 30% de los participantes respondieron que podrían distinguir si una voz fue generada por Inteligencia artificial o si era real. Además, alrededor del 10% de las personas encuestadas afirmó haber sido víctima de estafas de este tipo, perdiendo dinero en la mayoría de los casos [8].

Dado que menos de la mitad de los participantes en la encuesta contestaron que podrían identificar una voz generada por inteligencia artificial, es crucial determinar encontrar si esto es cierto y cuáles serían los mejores métodos para ayudar en este proceso. Si bien ya existen herramientas basadas en redes neuronales para identificar voces generadas por computadora, como Murf AI y LOVO [9], no existe una forma masiva y accesible de validar si estas soluciones son efectivas. Además, no se puede saber si las redes neuronales representan la forma más precisa o eficiente en cuanto a consumo de recursos para solucionar este problema. Incluso tanto Microsoft como OpenAI están desarrollando sus propias herramientas de clonación de voz, lo cual indica que el problema de imitación de voces solo seguirá incrementando [10].

Este estudio pretende llenar el vacío al comparar tres enfoques para la identificación de voces generadas: modelos de *machine learning* con filtros estadísticos aplicados, redes neuronales *Long Short-Term Memory* y el reconocimiento humano. Esta comparación no solo permitiría determinar cuál de los métodos es el más preciso, sino también evaluar la eficiencia en cuanto a consumo de recursos, lo cual podría ser fundamental para su aplicación práctica. Con esta información se podría establecer una solución más viable que pueda ser implementada en un entorno empresarial o un servicio para el publico general.

Aunque el reconocimiento humano tenga una menor precisión en la identificación de voces, incluirlo en la comparación podría aportar información que podría ser útil para la mejora de los otros modelos. Además, los resultados de este estudio podrían ofrecer una perspectiva acerca de la capacidad de las personas para detectar las estafas mencionadas anteriormente sin ningún tipo de asistencia tecnológica. Esto podría ser útil para diseñar intervenciones educativas y capacitar a la población para que puedan sobrellevar estos riesgos.

Por lo tanto, comparar diferentes métodos de identificación de voces generadas podría establecer una línea base en el ámbito de reconocimiento de voces o al menos servir como una opción viable dependiendo de los recursos a disposición. Incluso, al obtener las características respectivas de los audios se podría saber cuáles son las que más aportan a la decisión de los modelos y procesar datos futuros conforme a lo encontrado. Además, al comparar estos métodos con el más utilizado actualmente, redes neuronales, se podría determinar el mejor plan para casos específicos o para las necesidades de una empresa o situación. Los hallazgos podrían tener aplicaciones futuras en contextos donde la veracidad de la voz sea importante, ya sea en llamadas generales o sistemas de autenticación. Además, el estudio no solo aportaría al campo de ciencias de la computación al proveer métodos viables para la identificación de voces generadas, sino que también aportaría a la seguridad de la sociedad al mitigar el problema de voces clonadas.

III. Objetivos

A. General

Determinar el método más preciso para distinguir voces reales e imitaciones generadas por IA entre filtros estadísticos, redes Neuronales y la identificación humana.

B. Específicos

1. Determinar el método más eficiente en cuanto a utilización de recursos, tiempo y dinero.
2. Determinar si las personas mejoran en su identificación de voces generadas a medida que escuchan más de las mismas.
3. Encontrar los parámetros y estructura que funciona mejor para que una red neuronal tenga la mayor precisión en identificación.
4. Encontrar en el método filtros estadísticos las características obtenidas a través de *feature extraction* que más afectan a la precisión del modelo.
5. Encontrar los parámetros que logran que un filtro estadístico tenga la mayor precisión en la identificación de voces en la parte de *machine learning*.

IV. Marco teórico

A. Inteligencia Artificial

La inteligencia artificial es una rama de la computación y matemáticas que buscar recrear o imitar la inteligencia racional del ser humano. Sus aplicaciones se extienden a campos diversos, desde medicina hasta generación de contenido visual. Sin embargo, el uso de herramientas sin alguna supervisión o restricción puede presentar desafíos éticos que deben ser abordados.

1. Inteligencia Artificial

La inteligencia artificial, dependiendo del autor, puede tener diferentes definiciones. Mientras que algunos se basan en la habilidad de una computadora para imitar el pensamiento humano, otros definen este tipo de inteligencia a través de la racionalidad o la capacidad del programa para pueda realizar una acción "correcta" [11]. El término de "acción correcta" se vuelve abstracto en estos casos ya que depende del contexto donde se aplique. Es decir que el creador del algoritmo o programa decide cual es el resultado esperado que considera correcto o, al menos, puede dar una aproximación de lo que se considera correcto.

Esta rama de la computación no solo une las matemáticas y la estadística, sino que también se requiere una comprensión profunda del comportamiento humano, lo cual involucra al campo de la psicología. Aunque los distintos grupos tengan sus diferencias en cuanto a sus definiciones, al final terminan ayudándose entre sí, compartiendo ideas para fortalecer esta área de la computación. [11]

Aunque la inteligencia artificial es un término relativamente nuevo, aun así, se tiene una idea general de lo que implica. A grandes rasgos, la inteligencia artificial se refiere a la habilidad que tiene un programa de interpretar datos, aprender de los mismo y lograr que esos aprendizajes le sirvan para llevar a cabo una tarea de manera exitosa [12]. Sin embargo, cabe recalcar que este término no solo aplica a un área en específico, sino que puede cubrir muchos diferentes temas y se puede utilizar en innumerables contextos.

2. Aplicaciones de la Inteligencia Artificial

Dado que la inteligencia artificial utiliza cualquier tipo de dato para llevar a cabo sus tareas, esto permite una gran variedad de aplicaciones en diferentes campos. Existen diversas herramientas o aplicaciones que utilizan inteligencia artificial, ya sea para facilitar la vida o para mejorar la precisión de algún evento en específico. Entre sus aplicaciones

pueden destacarse los programas que se utilizan ya sea para medicina, educación o incluso entretenimiento.

En el ámbito médico existen diferentes campos en donde se puede aplicar la inteligencia artificial, desde revisión de tomografías hasta control de cirugías con robots. Aunque esta tecnología tiene un gran potencial para generar innovaciones, aún enfrenta desafíos relacionados con la disponibilidad y privacidad de los datos de los pacientes. A pesar de las restricciones que existan, estas herramientas han marcado un cambio significativo en la vida de los pacientes y los médicos. [13]

Aunque este tipo de aplicaciones sean muy importantes para una sociedad, la inteligencia artificial tiene como uno de sus objetivos facilitarles la vida a las personas. Esto se puede dar a través de asistentes de voz como Siri o Alexa, o incluso como un programa que recomiende canciones. Por otro lado, también se pueden encontrar aplicaciones en el ámbito creativo, las cuales sirven para la creación de imágenes, corrección de fotografías o incluso para la creación de logotipos. Sin embargo, su impacto va más allá de estas comodidades cotidianas, atención al cliente o expresión artística. Las tecnologías de procesamiento de lenguaje natural, como ChatGPT, están transformando la forma en la que la sociedad utiliza la inteligencia artificial.

3. Popularidad de la Inteligencia Artificial

La introducción de agentes de procesamiento de lenguaje natural como lo es ChatGPT, ha impulsado la popularidad de la inteligencia artificial. Esta tendencia se aceleró aún más durante la pandemia de COVID-19, la cual motivó la adopción de nuevas tecnologías debido a la necesidad de comunicación remota. Para adaptarse a las nuevas dificultades, muchas compañías empezaron a utilizar el método de trabajo remoto mientras que otras personas aprovecharon este tiempo para iniciar un negocio en línea. Dado que muchos de estos negocios solo eran gestionados por una persona, se vieron obligados a implementar automatizaciones que mejoraran la experiencia del cliente. Estas automatizaciones incluían *chatbots* para responder a los clientes o generar contenido de forma rápida y eficiente. [14]

Como derivado de los *chatbots*, se obtuvo la herramienta de ChatGPT, la cual no solo puede procesar lenguaje natural, sino que sobre todo se utiliza como generador de texto, lo cual lo convierte en una herramienta valiosa para este tipo de negocios. Puede ser utilizado para generar contenido para el comercio, para aprender sobre estrategias empresariales e incluso entrenarlo para que funcionase como guía para las políticas de la empresa. [15]

El aprendizaje y generación de contenido fue una de las razones por las que se volvió tan popular esta tecnología. A medida que más personas empezaban a utilizar esta aplicación, se descubrieron más formas de aprovecharla. Debido a la gran cantidad de información que fue usada para entrenar a ChatGPT, estudiantes empezaron a utilizarlo como ayuda en sus tareas, ya fuera para escribir un texto, para estudiar, actuar como tutor, para cualquier cosa que se les ocurriese. La adopción de la herramienta por parte de este grupo de personas fue una de las principales razones de éxito de ChatGPT. [16]

Sin embargo, el lanzamiento de esta tecnología impulso el desarrollo de una variedad de otras herramientas que también utilizarían la inteligencia para generar sus productos. Empresas y desarrolladores vieron el éxito y las capacidades de ChatGPT como una oportunidad para innovar y crear nuevas aplicaciones que aprovechan el poder de los modelos de lenguaje grandes. Esto llevó a la creación de asistentes virtuales, generadores de contenido automatizado, herramientas de análisis de datos, Generador de imágenes, e incluso imitadores de voces. [17]

4. Peligros de la Inteligencia artificial

A pesar de los grandes avances y beneficios que ofrece la inteligencia artificial, esta también presenta ciertos desafíos y peligros para la sociedad. Uno de los problemas más grandes para un modelo de inteligencia artificial es el sesgo. Debido a la gran cantidad de datos que se utilizan para entrenar estos modelos, es posible que dichos datos contengan sesgos, lo que puede llevar al modelo a producir resultados "incorrectos". En este contexto el término “incorrecto” se refiere a resultados inesperados o no deseados. En estos casos, donde los modelos se ven sesgados, puede conducir a que realicen decisiones que puedan ser perjudiciales para cierto grupo de personas no representado por los datos estudiados. [18]

Además, existen herramientas de inteligencia artificial el cual su único propósito es imitar. Sus aplicaciones se enfocan en replicar estilos de escritura, formas de pintar y otros patrones que hayan encontrado en el conjunto de datos para su entrenamiento. Sin embargo, este tipo de imitaciones puede generar problemas graves, como la violación de derechos de autor. Mas preocupante aún, esta tecnología también se puede utilizar para imitar las voces y caras de otras personas, en algo llamado *deepfake*. Este término se refiere a la modificación de imágenes o videos para simular las facciones, voz o comportamiento de alguna persona. Estos *deepfakes* se han utilizado para difundir información falsa que pueda dañar la imagen de una persona y llevar a cabo estafas, lo cual incrementa su nivel de peligro para la sociedad.

Dado el alto nivel de realismo que presentan estas imitaciones, resulta difícil distinguir si algo fue en realidad dicho o hecho por una persona. Por ello, los *deepfakes* se han popularizado en las redes sociales para difundir la imagen equivocada de alguien o para tomar crédito por cosas ajenas. Esta tecnología se ha vuelto muy popular sobre todo para llevar a cabo ataques de ingeniería social. [19]

B. Redes Neuronales

Las redes neuronales y los modelos de *machine learning* son herramientas fundamentales para el ámbito de la inteligencia artificial. Estos algoritmos permiten que las máquinas reconozcan patrones en conjuntos de datos y realizar predicciones en base a lo aprendido.

1. Tipos de redes neuronales y su uso

Las redes neuronales son un sistema de computación diseñado para imitar la forma en que funcionan los cerebros, utilizando estructuras llamadas “neuronas”. Estas redes se

componen de capas de neuronas conectadas entre sí, donde cada conexión tiene un "peso" ajusta durante el entrenamiento para minimizar el error en las predicciones. El peso se refiere a que tanto aporta cada neurona hacia al resultado final, permitiendo que la red aprenda a tomar decisiones en base a patrones complejos presentes en los datos de entrada.

Las redes neuronales se utilizan en una amplia gama de aplicaciones, desde el reconocimiento de imágenes y la traducción de idiomas hasta la detección de fraudes y la conducción autónoma. Dependiendo de su aplicación puede que varie la arquitectura de la red y el diseño de las neuronas. A continuación, se describen algunos de los tipos de redes neuronales y su uso.

Las redes neuronales convolucionales (CNN) son útiles para procesar datos que vienen en forma de cuadrícula, como las imágenes. Estas redes aplican filtros a los datos de entrada para extraer características relevantes, como bordes, texturas y patrones. Estas características se organizan en "mapas de características" que son procesados por capas adicionales de la red para identificar estructuras más complejas y abstractas. Cada capa de la red aprende a detectar diferentes características a medida que las imágenes pasan a través de la red, lo que permite a las CNN construir una representación jerárquica de las imágenes. Esta jerarquía es representada al encontrar las características más simples en capas tempranas de la red, mientras que las más complejas se van construyendo poco a poco con combinaciones de lo que encontraron capas anteriores. De esta manera, se pueden reconocer más patrones que puedan parecer más complejos y llegar a la meta de identificar lo que se desea. [20]

Por otro lado, se encuentran las redes generativas antagónicas (GANs), se usan para generar contenido o datos. Esta red neuronal combina dos redes neuronales que compiten entre sí. Una de las redes, la red generadora, se entrena para que pueda generar contenido parecido al que se tiene como datos de entrenamiento. Mientras que la otra, la red discriminadora, trata de discernir si el contenido generado fue creado artificialmente o bien, pertenece a algún dato real. El propósito es lograr que la red generadora "gane", y esto se logra al engañar a la red discriminadora dentro de un margen aceptable. En ese punto, se puede utilizar la red generadora para la creación realistas similares a los vistos durante el entrenamiento. Este tipo de redes pueden utilizarse en tareas de creación de imágenes, sonidos y otro tipo de contenido como texto. [21]

Finalmente, las redes neuronales recurrentes (RNN) están diseñadas para procesar datos secuenciales, como series de tiempo o lenguaje natural, ya que mantienen una "memoria" de lo procesado anteriormente. Esto las hace útiles para tareas donde es importante considerar información del pasado o bien donde sea necesario un contexto como el reconocimiento de voz. Dado que la voz es un sonido, se puede interpretar como una onda a lo largo del tiempo. La arquitectura de una RNN es similar a la de una red *feed-forward,* con la diferencia que, ahora las neuronas también utilizan como entrada el estado anterior o la salida de la misma neurona en el paso de tiempo previo. Sin embargo, las RNN enfrentan un desafío conocido como el "desvanecimiento" o "explosión" de gradiente. Este término hace referencia a los pesos de la red, los cuales, al actualizarse, presentan un cambio tan pequeño o grande que la red no logra aprender. Para mitigar este problema se

desarrollaron variantes como las redes *Long Short-Term Memory* (LSTM), las cuales introducen mecanismos específicos para manejar mejor la información secuencial. [22]

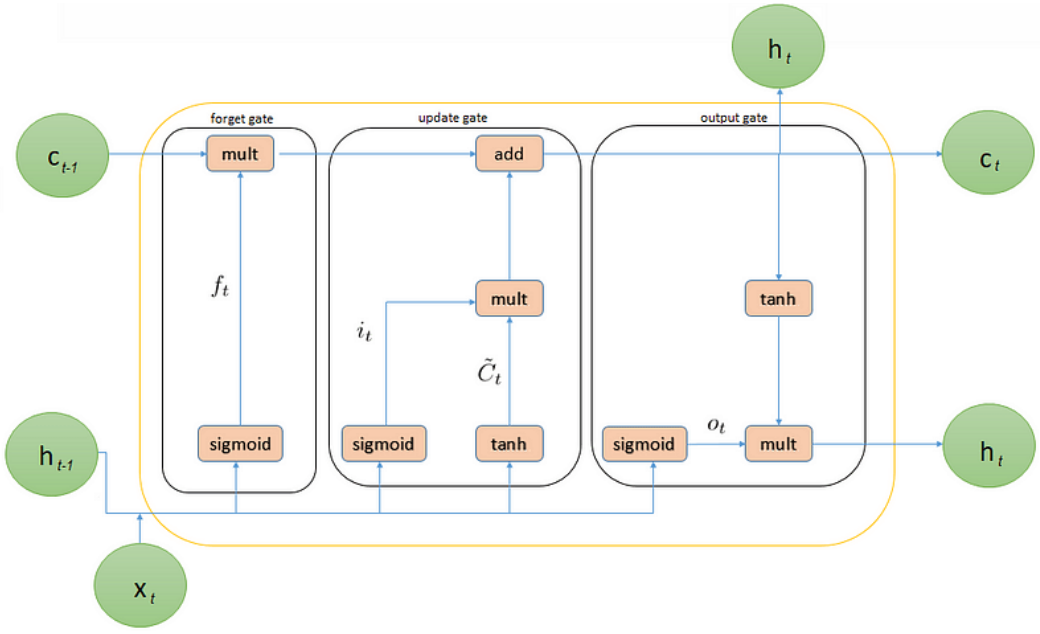
2. Red Neuronal de memoria de corto-largo plazo (LSTM)

Las redes neuronales *Long Short-Term Memory* son una variante de las redes neuronales recurrentes. A comparación de estas redes, esta nueva arquitectura puede retener información más antigua en una secuencia temporal. Por esa misma razón, es útil en tareas como el reconocimiento de voz, escritura y el modelado de música.

Aunque parten de la estructura de las RNN, esta es una arquitectura mucho más compleja, capaces de superar el problema del gradiente que desvanece o explota. Esta red implementa una memoria especial que le permite administrar la información que guarda a lo largo del tiempo. Como se puede observar en la Figura 1, cada neurona de la red contiene una celda de memoria que guarda información a largo plazo, junto con tres "puertas", la puerta de entrada, puerta de olvido y puerta de salida.

La puerta de entrada controla la cantidad de información que puede entrar a la celda de la memoria. A través de una función de activación, la puerta determina el grado de relevancia de la información. La puerta de olvido decide que tanta información ya no es relevante para el modelo y la descarta. Por último, la puerta de salida decide que tanta información se utilizará para generar el resultado final de la neurona en ese momento de tiempo.

Figura 1: Arquitectura de una neurona LSTM



Fuente: Towards AI [23]

El proceso de una red LSTM empieza cuando recibe un dato en un momento especifico. Esa información es recibida por la puerta de entrada y esta decide la información que es

relevante para llevar a la celda de memoria. Al mismo tiempo, la puerta del olvido decide que tanta de la información ahora almacenada debe ser olvidada y retirada de memoria. Una vez se tengan esos dos resultados, se actualiza la memoria con los cambios hechos por tanto la puerta de olvido como la de entrada. Por último, la puerta de salida decide que tanta información de la memoria es relevante para el resultado final. [24]

Es importante notar que esta es la arquitectura estándar de una red LSTM. Existen variantes de esta red, sin embargo, en un estudio [25] realizado se observó que, de las diferentes variantes estudiadas, ninguna aporta una mejora significativa a la arquitectura estándar. Esos fueron los resultados vistos al utilizar este tipo de red y sus variantes en problemas de reconocimiento de escritura a mano y modelado de música.

3. Utilización de Redes Neuronales para la identificación de voces *deepfake*

Las redes neuronales han demostrado ser herramientas eficaces en la detección de voces generadas por inteligencia artificial (IA), también conocidas como *deepfakes*. Este campo ha cobrado relevancia debido a los avances en la síntesis de voz, que pueden producir grabaciones altamente realistas y difíciles de distinguir de las voces humanas reales.

Las redes neuronales LSTM son un tipo de arquitectura adecuado para procesar señales de audio, dado que pueden guardar información relevante a lo largo de series de tiempo extensas. Esto es crucial en la detección de *deepfakes* de voz, donde las características sintéticas pueden ser sutiles [26]. En un estudio [27] realizado se utilizó una variante de la LSTM llamada LSTM bidireccional, que incorpora información tanto del pasado como del futuro en un tiempo determinado, lo cual permite un análisis más completo. Esto fue posible debido a la extracción de características espectrales para momentos específicos a lo largo del audio.

Por otro lado, también se pueden utilizar redes neuronales convolucionales para la detección de voces *deepfake.* Las CNN son especialmente eficaces en la extracción de características espaciales de los datos de audio, como los espectrogramas, lo que les permite identificar patrones sutiles que pueden indicar si una voz ha sido generada por IA. Al analizar los espectrogramas de las señales de audio, las CNN pueden aprender a diferenciar entre las voces humanas auténticas y las sintetizadas por IA mediante la detección de anomalías y características específicas que son difíciles de replicar para los modelos generativos. [26]

Además, existen modelos de redes que juntan tanto una red neuronal CNN como una red LSTM para una mayor precisión. Estos modelos híbridos aprovechan las fortalezas de ambas arquitecturas: las CNN son eficaces en la extracción de características espaciales de los espectrogramas de audio, mientras que las LSTM son adecuadas para capturar dependencias temporales en las secuencias de audio. Al utilizar una CNN para extraer características detalladas de los espectrogramas y luego procesar estas características con una LSTM para analizar las relaciones temporales, los modelos pueden identificar patrones complejos y sutiles que son indicativos de voces generadas por IA. Esta combinación permite una detección más precisa, mejorando significativamente la capacidad de los

sistemas para distinguir entre voces humanas auténticas y voces generadas por inteligencia artificial. [28]

C. Filtros de audio

La detección de voces generadas por inteligencia artificial (IA) requiere técnicas avanzadas de procesamiento de audio donde los filtros de sonido aportan mucho. Al aplicar filtros específicos, es posible resaltar características del audio que permiten distinguir ciertos patrones diferentes entre voces reales y *deepfake.* No obstante, para entender su aplicación en reconocimiento de voz es importante conocer las características de audio que juegan un rol en esta tarea.

1. Características de una señal de audio

Las características de una señal de audio son esenciales para su análisis, procesamiento y manipulación. Estas características permiten describir y entender la naturaleza del sonido, facilitando diversas aplicaciones como la síntesis de voz, reconocimiento de patrones y detección de voces generadas por inteligencia artificial.

El espectrograma es una representación visual de todas las frecuencias presentes en una señal de audio a lo largo del tiempo. Se obtiene mediante la aplicación de la transformada de Fourier a segmentos de la señal, mostrando la distribución de energía en función del tiempo y la frecuencia. Dado que presentan mucha información sobre la señal de audio, las demás características son un derivado pues tan solo representan una parte de todo el espectro del audio. Además, al ser una representación visual, también se utiliza en redes neuronales convolucionales para un análisis profundo del audio.

Partiendo del hecho que el espectrograma incluye información sobre el espectro de frecuencias en un audio, la frecuencia de una señal de audio se refiere al número de ciclos que una onda sonora completa por segundo, medida en Hertz (Hz). La frecuencia determina el tono del sonido: frecuencias más altas producen tonos más agudos, mientras que frecuencias más bajas resultan en tonos graves. [29].

Por otro lado, la amplitud de una señal de audio indica la intensidad o el volumen del sonido. Se representa visualmente como la altura de las ondas en un gráfico de forma de onda. La amplitud se relaciona directamente con la percepción del volumen: mayores amplitudes corresponden a sonidos más fuertes, mientras que menores amplitudes producen sonidos más bajos. Niveles excesivamente altos de amplitud pueden causar una distorsión la percepción del sonido, afectando la calidad de este. [30]

El timbre es la cualidad del sonido que permite distinguir diferentes fuentes sonoras, incluso si tienen la misma frecuencia y amplitud. Está determinado por la forma de la onda sonora y los armónicos presentes, que son frecuencias adicionales que acompañan a la frecuencia fundamental o bien la frecuencia natural. El timbre permite diferenciar un sonido de otro pese a que se hayan tocado con la misma intensidad, es decir que tengan la misma amplitud y tono [31].

La duración de una señal de audio se refiere al tiempo durante el cual se mantiene un sonido, mientras que el ritmo es el patrón temporal de los sonidos en una secuencia. La duración y el ritmo son esenciales en música y habla, ya que influyen en la estructura y la comprensión de las secuencias de sonido [32].

La característica de sonido MFCC es una representación del espectro de potencia de una señal de audio, basada en una escala de frecuencia Mel, que asemeja más a la percepción humana del sonido. Se utilizan ampliamente en el reconocimiento de voz y análisis de audio. Los MFCC se calculan a partir del espectrograma y capturan pequeñas características, normalmente entre 10 a 20 las cuales describan el espectro de frecuencias. [33]

Por otro lado, también existen los LFCC los cuales funcionan casi igual que los coeficientes MFCC, con la diferencia que estos no utilizan la frecuencia Mel, sino que utilizan una frecuencia lineal. Este cambio los hace más susceptibles a frecuencias altas, lo cual puede ser útil cuando se están procesando voces más agudas o se intentan encontrar diferentes patrones en el sonido [34].

También existen las características de sonido Coeficientes Cepstrales Constantes-Q (CQCC), estos permiten tener más resolución acerca de las diferentes frecuencias para las frecuencias bajas, mientras que para las altas permiten observar mejor que cambios se han dado en la señal. En otras palabras, para las frecuencias bajas permite diferenciar frecuencias que estén una muy junta de la otra y para las frecuencias altas permite observar cambios en la señal en intervalos de tiempo muy cortos. Por esto mismo, estos coeficientes han demostrado ser efectivos en tareas de detección de *deepfake* de voz y otras aplicaciones de seguridad del habla [35].

El *Zero-Crossing Rate* ZCR mide la frecuencia con la que la señal de audio cruza el eje cero, es decir, cambia de positivo a negativo o viceversa. En tareas de verificación de voz, sirve para diferenciar entre el audio de las voces y el audio de ruido de ambiente. Un ZCR alto indica que existen muchas variaciones en el sonido en las cuales cambia de positivo a negativo en la escala, mientras que uno bajo indica que es un sonido más continuo [36].

Los formantes son picos en el espectro de frecuencia de la voz humana que corresponden a resonancias en el tracto vocal. Estos picos son muy útiles para reconocer sonidos de consonantes o vocales cuando una persona habla. Debido a esta cualidad, los formantes permiten identificar patrones particulares en el habla, facilitando la distinción entre diferentes sonidos en el audio. Esto resulta útil en aplicaciones de reconocimiento de voz pues permite clasificar la calidad y el tipo de habla encontrado en el audio. [37]

2. Filtros de sonido

Los filtros de sonido son herramientas esenciales en el procesamiento y análisis de señales de audio. Permiten la manipulación de las frecuencias presentes en una señal para mejorar la calidad del audio, eliminar ruido o extraer características relevantes. A continuación, se describen los tipos principales de filtros de sonido y estadísticos al igual que sus aplicaciones.

Entre los filtros más comunes se encuentran los filtros *low-pass* y *high-pass,* los cuales atenúan frecuencias bajas y altas respectivamente. Los filtros *low-pass* permiten el paso de frecuencias por debajo de un umbral específico y atenúan las frecuencias superiores a ese umbral. Son útiles para eliminar ruidos de alta frecuencia y suavizar señales. Se utilizan en sistemas de audio para reducir el ruido de alta frecuencia. Además, son utilizados muy comúnmente para eliminar el ruido de alta frecuencias en grabaciones de voz.

En contraste con los filtros *low-pass*, los filtros *high-pass* permiten el paso de frecuencias por encima de un umbral indicado y atenúan las frecuencias inferiores a ese umbral. Son útiles para eliminar ruidos de baja frecuencia, como el ruido de fondo de la red eléctrica. Se emplean en audio para mejorar la claridad del sonido eliminando frecuencias bajas indeseadas [38].

Los filtros *Band-Pass*, combinan las características de los anteriores filtros, permiten el paso de un rango específico de frecuencias y atenúan las frecuencias fuera de ese rango. Son útiles para seleccionar un rango específico de frecuencias para su análisis o procesamiento. Se utilizan en aplicaciones de telecomunicaciones para filtrar señales dentro de una banda de frecuencia específica [36].

Los filtros de rechazo de banda, también conocidos como filtros *notch*, funcionan de manera opuesta a los filtros *Band-Pass*. Este tipo de filtro atenúan un rango especifico de frecuencias y permiten el paso de las demás. En otras palabras, atenúan las frecuencias intermedias mientras que dejan pasar las que se encuentran en los extremos. Son útiles para eliminar interferencias de frecuencias conocidas, como las señales de 50/60 Hz, que suelen introducir ruidos no deseados en las grabaciones de audio [32].

Dentro de los filtros basados en datos estadísticos, se puede encontrar el filtro de media móvil y también el filtro de mediana. El filtro de media es uno de los filtros más comunes para procesamiento de señales. El punto de este filtro es suavizar la señal, eliminando fluctuaciones que pueden ser consideradas como ruido. Sin embargo, su desventaja es que también puede atenuar características importantes de la señal, como transiciones rápidas en frecuencia. Este filtro utiliza una ventana de tiempo para ir calculando la media de ese rango de tiempos y asignarle el valor resultante al dato que se está observando. Debido a esto, las ventanas se realizan con número impares, para que el dato observado siempre tenga la misma cantidad de “vecinos” de cada lado. [39]

El filtro de mediana funciona de manera parecida al filtro de medias, sin embargo, ahora se calcula la mediana y ese es el valor que se le asigna. Este tipo de filtros es eficaz para eliminar ruido, donde se encuentren picos agudos en la señal. Al reemplazar el valor de la señal por el valor mediano de sus vecinos, el filtro logra preservar detalles importantes de la señal, lo cual lo hace útil en aplicaciones donde la calidad del sonido es importante. [40]

D. Audición humana

La audición humana permite percibir y analizar sonidos en el entorno, desempeñando un papel indiscutible en la comunicación e interpretación del mundo. Por lo general, el sistema auditivo es sensible cambios en el tono y frecuencia en los sonidos, lo cual permite

distinguir entre variaciones sutiles en la música, el habla y otros sonidos ambientales. Esta capacidad de identificar estos cambios sirve para la detección de patrones incluyendo la forma de hablar de las personas.

1. Tipos de sonidos detectables por los humanos

La audición humana es un proceso complejo que permite a las personas detectar, identificar y diferenciar una amplia gama de sonidos. Este proceso no solo implica la captación de las ondas sonoras, sino también su interpretación en el cerebro. La capacidad humana para percibir sonidos abarca varios aspectos, como la detección de diferentes tipos de sonidos, la sensibilidad a los cambios de frecuencia y tono, y la capacidad de identificar pequeños cambios en los sonidos.

Los humanos pueden detectar sonidos en un rango de frecuencia que va desde los 20 Hz hasta 20,000 Hz. Los sonidos de baja frecuencia, por debajo de 250 Hz, incluyen ruidos graves como el retumbo de un trueno, el zumbido de un motor grande o el rugido de un león. Este tipo de sonidos a menudo se sienten tanto como se escuchan, ya que las ondas sonoras largas pueden hacer vibrar objetos y estructuras.

Por otro lado, los sonidos de alta frecuencia, por encima de 8,000 Hz, incluyen tonos agudos como el canto de un pájaro, el chirrido de un insecto y ciertos componentes de la voz humana, como los sonidos de las consonantes "s" y "t". Los sonidos de alta frecuencia son más direccionales y tienden a atenuarse más rápidamente con la distancia que los sonidos de baja frecuencia. A medida que las personas envejecen, su capacidad para escuchar sonidos de alta frecuencia generalmente disminuye, una condición conocida como presbiacusia. [41]

En general, el rango de frecuencia más sensible para los humanos está entre 1,000 Hz y 4,000 Hz, que es donde se encuentra la mayoría de los sonidos del habla. Este rango es crucial para la comunicación efectiva, ya que permite la detección y comprensión de las vocales y consonantes que conforman el habla. Las frecuencias dentro de este rango son especialmente importantes para distinguir diferentes fonemas y para comprender el tono emocional de la voz de una persona. Los formantes, los cuales son esenciales para la identificación de diferentes vocales y consonantes, también caen dentro de esto rango, y su percepción correcta permite la comprensión clara del habla. Además, la percepción de las transiciones rápidas entre diferentes sonidos del habla (consonantes y vocales) depende de la capacidad auditiva en este rango de frecuencia. [42]

Al igual que en el habla, la percepción de frecuencias también es fundamental en la música, la cual abarca una amplia gama de frecuencias y tonalidades, desde los tonos graves del contrabajo hasta los agudos del violín. La percepción musical va más allá de la detección de las notas, sino también implica la capacidad de identificar variaciones en el timbre y ritmo. El timbre, que es la calidad del sonido que permite distinguir entre diferentes instrumentos tocando la misma nota, depende de la percepción de las frecuencias armónicas que acompañan a la frecuencia fundamental de una nota. [43]

2. Sensibilidad a los cambios de frecuencia y tono

La capacidad del oído humano para diferenciar entre dos tonos con frecuencias cercanas se conoce como resolución de frecuencia. Los humanos tienen una resolución de frecuencia muy alta, logrando detectar diferencias de frecuencia de tan solo unos pocos Hertz, especialmente en el rango medio donde la sensibilidad es mayor [44].

Además, los humanos también cuentan con la percepción de tono, la cual está relacionada con la frecuencia del sonido. Los humanos pueden identificar pequeños cambios en el tono, lo cual es esencial para la música y el habla. La discriminación de tono se refiere a la capacidad de distinguir diferencias en la altura del sonido, y es especialmente aguda en el rango de frecuencias medias [45].

3. Identificación de cambios sutiles en un audio

La capacidad humana para detectar y analizar sonidos es un proceso complejo y altamente refinado que involucra múltiples aspectos del sistema auditivo. Esta habilidad no solo permite a los humanos comunicarse de manera efectiva a través del habla, sino que también juega un papel crucial en la percepción de la música y en la interpretación de diversos sonidos del entorno. Entre las muchas características del sonido que los humanos pueden detectar, los cambios sutiles en la amplitud, el timbre y ritmo son muy importantes. Estos cambios proporcionan información esencial que ayuda a los individuos a entender mejor su entorno auditivo y a identificar la fuente de los sonidos. A continuación, se explorarán en detalle estas capacidades auditivas específicas y su importancia en la vida diaria.

Los humanos pueden detectar cambios en la amplitud del sonido, lo que se percibe como variaciones en el volumen. Esta capacidad es fundamental para entender el habla y la música, permitiendo identificar cuándo una persona está hablando más fuerte o débil, y captar las sutilezas de las piezas musicales. La detección de cambios en la amplitud también es crucial para la localización de la fuente del sonido en un entorno. Cuando un sonido es más fuerte, generalmente percibimos que está más cerca, y cuando es más débil, percibimos que está más lejos. [46]

Por otro lado, el timbre de un sonido proporciona mucha información para el cerebro humano. Esta característica es esencial para la identificación de diferentes instrumentos musicales y voces. Los humanos son extremadamente sensibles a los cambios en el timbre, lo cual es crucial para reconocer las voces de diferentes personas, incluso si están hablando en el mismo tono y volumen. Esta sensibilidad al timbre también permite a los humanos distinguir entre diferentes tipos de sonidos que pueden tener la misma frecuencia y amplitud. [43]

Finalmente, ritmo del sonido aporta información temporal del sonido para el ser humano. Esta capacidad es muy importante para la comprensión del habla, donde el ritmo y las pausas juegan roles importantes en la comunicación. Los humanos pueden discernir la duración de las sílabas y las palabras, lo cual es esencial para la comprensión del lenguaje hablado. El flujo y la inflexión del habla, también proporciona información importante

sobre el estado emocional y la intención del hablante. Por ejemplo, una pausa prolongada puede indicar reflexión o duda, mientras que un ritmo rápido puede transmitir felicidad o urgencia. Esta sensibilidad temporal permite a los humanos procesar y reaccionar rápidamente a los cambios en su entorno auditivo [41].

V. Metodología

El objetivo principal de esta investigación fue determinar el método más preciso entre tres enfoques para identificar si una voz había sido generada por inteligencia artificial. Para ello, se implementaron distintas etapas, que incluyeron la recolección de datos, la limpieza de los datos, la creación de modelos y del examen para el reconocimiento humano, el ajuste de hiperparámetros y la comparación de resultados. Cada una de estas fases contribuyó a obtener resultados concluyentes sobre el rendimiento y eficiencia de los métodos comparados.

A. Recolección de los datos para modelos los de *machine learning*

Los datos utilizados en este estudio se recuperaron de la base de datos ASVspoof 2021 [47], la cual proporciona audios con voces reales como generados con diversos algoritmos de IA. Como se muestra en el Cuadro 1, los audios estaban categorizados en distintos conjuntos de datos: PA (Physical Access), LA (Logical Access), y DF (Speech Deepfake). Este último conjunto era similar al de LA, pero sin la verificación de la identidad de la persona que hablaba. Para este estudio, se utilizó el conjunto de datos DF, ya que el objetivo era determinar si la voz había sido generada por inteligencia artificial o no.

Cuadro 1: Características de los conjuntos de datos disponibles en ASVspoof Challenge 2021

Conjunto de datos	Descripción	Tipo de voz	Tecnología aplicada	Calidad de grabación	formato del audio
PA (*Physical Access*)	Grabaciones en espacios físicos	Voces reales/manipuladas	Grabadoras de diversas calidades	Variable	flac
La (*Logical Access*)	Audios originales y manipulados por IA junto verificación de identidad.	Voces reales y sintéticas	TTS, VC y redes telefónicas	Alta / moderada	flac
DF (*Deepfake*)	Audios originales y manipulados por IA.	Voces reales y sintéticas	TTS y VC	Alta / moderada	flac

Este conjunto de datos consistía en 611,829 audios, divididos entre voces reales (*bonafide*) y voces generadas por inteligencia artificial (*deepfake*). Además de los audios, existían otros conjuntos de datos complementarios que definían etiquetas para cada uno de los audios, las cuales se utilizaron para las tareas de preprocesamiento.

B. Preprocesamiento de los datos:

Para el preprocesamiento de los datos se utilizó Python, junto con las librerías de Numpy, Matplotlib, Mutagen, Pandas, Scikit-Learn, Librosa y Soundfile. Se inició con la extracción de metadatos relevantes de cada archivo, como su duración*, bitrate*, *sample rate, bit depth* y número canales. Una vez obtenidos estos datos, se integraron en un solo *DataFrame* junto con las etiquetas de los audios.

Una vez obtenido el *DataFrame* con todos los datos necesarios, se calcularon estadísticas importantes para comprender mejor el conjunto de datos. Por ejemplo, se obtuvo la distribución de audios con voces reales y generadas por inteligencia artificial. Además, se analizó la distribución de la duración de los audios a lo largo del conjunto de datos para las ambas clasificaciones. Para lograr obtener estadísticas significativas sobre los datos se utilizó Pandas junto con Numpy, mientras que para visualizar los datos se utilizó Matplotlib.

Después de finalizar el análisis exploratorio de los datos, se procedió al procesamiento de estos para pudieran ser utilizados por los tres métodos estudiados. Esto comenzó con la limpieza de los datos, descartando todas aquellas características que no serían útiles para las siguientes etapas. Además, se estableció un estándar de duración entre 2 y 4 segundos para garantizar uniformidad y asegurar que los modelos recibieran datos consistentes. La razón de esta elección fue debido a que la mayoría de los datos se encontraban dentro de este rango de duración, además, al recortarlos no perderían tanta información. Todos los audios que cumplieran con esta condición se recortaron a una duración de 2 segundos, utilizando el inicio del audio como punto de partida. Este recorte permitió adaptar los datos al modelo de redes neuronales LSTM el cual requiere series de tiempo uniformes.

Como último paso del procesamiento, se extrajeron las características de los audios obtenidos del paso anterior, con la librería Librosa en Python. Dentro de las características extraídas se pueden encontrar, *zero-crossing rate, root-mean-square, specttral centroid, spectral bandwith, spectral rolloff, spectral contrast, Chroma Short-term Fourier Trasnformation (STFT) y* 10 coeficientes cepstrales en las frecuencias de Mel *(MFCC por sus siglas en inglés)*. Cada una de estas características fueron calculadas para un lapso llamado *frame*, dando un total de 87 *frames* por audio para cada una de las características. Es importante resaltar que debido a la gran cantidad de datos que se estaban guardando, fue necesario dividirlo por lotes e ir guardando lo obtenido de cada lote en archivos utilizando la librería Pickle en Python.

Al tener un archivo con todos los datos necesarios, faltaba dividirlos en conjuntos de entrenamiento, prueba y validación. Para ello, se utilizó la función "train_test_split de la librería Scikit-Learn en Python, con una partición de 0.2 para la parte de pruebas y 0.8 para la parte de entrenamiento. Es importante destacar que se tuvieron en cuenta dos aspectos: primero, las clases del conjunto de datos estaban desbalanceadas, existían más datos de

voces *deepfake* que de *bonafide*, por lo que se aseguró obtener el 20% de los datos de ambas categorías por separado y luego juntarlos para mantener una proporción equitativa; segundo, se utilizó una "semilla" para garantizar que los datos seleccionados por la función fueran siempre los mismos.

Por último, los datos de validación fueron extraídos de la data anteriormente dedicada para entrenamiento, con una proporción del 20% con respecto a esos datos. Este conjunto de datos fue útil para confirmar el rendimiento de los modelos ya que son datos nunca se utilizaron durante el entrenamiento, como lo dice su nombre, sirven para validar que no haya un sobreajuste para los datos de entrenamiento.

C. Creación de la encuesta:

Con el objetivo de determinar la capacidad de los humanos para identificar si una voz había sido generada o no por inteligencia artificial, se desarrolló un examen en la plataforma QuestionPro. Este examen consistió en 2 partes: la primera recopilaba información demográfica, como género, edad, profesión, mientras que la segunda parte era una prueba dicotómica, en la cual se le presentaba al participante con audio y debía escoger entre 2 opciones, "voz real" o "voz generada por inteligencia artificial".

Los audios para el examen fueron seleccionados aleatoriamente del conjunto de datos de prueba previamente separado. Se seleccionaron 10 audios en total (5 de cada categoría) y se utilizó función *shuffle* de la librería Random en Python para llevar a cabo un barajamiento con la función de establecer el orden en el que serían presentados los audios durante el examen.

El examen fue administrado en línea y se distribuyó a personas de distintas edades, logrando obtener una muestra de aproximadamente 95 participantes. Cabe mencionar que todos los participantes fueron informados sobre la naturaleza del estudio y dieron su consentimiento informado antes de participar. Además, el examen se realizó en anonimato, sin forma de identificar a una persona en base a las respuestas que haya dado. Al agrupar las respuestas en base a edad o profesión, se pudo analizar qué grupo de personas compartía características que ayudara a su rendimiento en identificar si una voz era real o generada por inteligencia artificial.

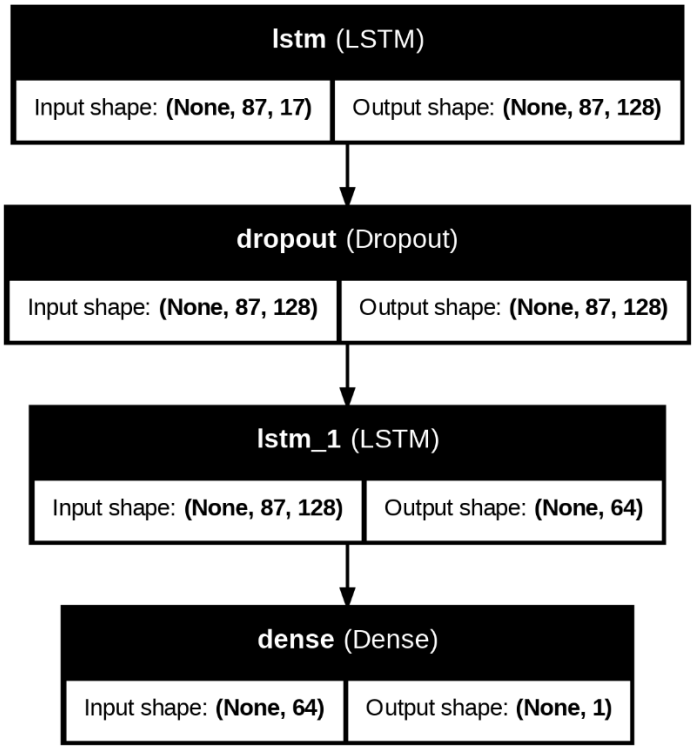
D. Creación y entrenamiento de los modelos de *machine learning*

Se desarrollaron dos enfoques diferentes para la creación y entrenamiento de modelos de machine learning: uno basado en la extracción de características utilizando una red neuronal de arquitectura *Long Short-Term Memory* (LSTM), y otro basado en diversos modelos de *machine* learning junto con la extracción de características, seguida de la aplicación de filtros estadísticos. Además de comparar el rendimiento de los modelos, también se analizó el consumo de recursos durante el entrenamiento y predicción.

Para el modelo LSTM, primero se definió la arquitectura de la red neuronal. Como se puede observar en la Figura 2, esto incluye la elección del número de capas LSTM y la cantidad de unidades en cada capa, lo que permitió capturar las dependencias temporales en las secuencias de audio. La entrada del modelo fue definida con un Numpy Array con la

forma (87, 17), donde cada audio se representaba con 87 *frames,* y cada *frame* contenía 17 datos, uno por cada característica. Posteriormente, los datos se estandarizaron utilizando *StandardScaler* de Scikit-learn. Después de haber creado las capas dos capas LSTM con una capa de *Dropout* entre ellas, se incluyó una capa final para procesar la salida y realizar la clasificación binaria final. La función de activación utilizada en la capa de salida fue una sigmoide, adecuada para clasificaciones binarias.

Figura 2: Arquitectura de la red neuronal LSTM



El modelo se compiló con el optimizador ADAM y la función de pérdida binaria *binary crossentropy*. Para el entrenamiento del modelo, se utilizó un entorno de Google Colab con acceso a una GPU NVIDIA Tesla T4. Por último, se implementó *Early Stopping* para detener el entrenamiento del modelo si el valor de la función de pérdida no mejoraba después de 3 épocas, con el objetivo de ahorrar recursos y evitar el sobreajuste del modelo.

Para la creación de los modelos de *machine learning* se utilizaron las librerías de Autogluon y también Scikit-Learn. Para ello, de primero se realizó un paso de procesamiento adicional, por cada uno de los audios, se obtuvo la media de todos los *frames* para cada una de las características. Luego, los datos fueron estandarizados usando *StandardScaler*, asegurando que tanto los datos de entrenamiento como los de prueba se ajustar a la misma escala. Con los datos preparados, se entrenaron todos los modelos con el conjunto de datos acorde, con los hiperparámetros por defecto de los modelos.

Posteriormente, se crearon conjuntos de datos aplicando distintos filtros estadísticos como: filtro *high-pass*, filtro *low-pass*, filtro *band-pass* filtro *notch*, filtro de media y filtro

de mediana. Al igual que con el entrenamiento inicial, el paso de estandarización también fue aplicado a estos conjuntos de datos. Los modelos previamente creados fueron utilizados para evaluar los nuevos conjuntos de datos sin ser reentrenados. Esto permitió observar cuales fueron los filtros que tenían mejor rendimiento y con los resultados obtenidos se podría indicar en que tipo de datos se enfocaban los modelos para sus predicciones.

Por último, se utilizó la librería de Autogluon para identificar la importancia de las diferentes características de audio extraídas en la predicción de los modelos. Esto permitió calcular la contribución de cada característica al rendimiento del modelo, lo cual ayudó a entender cuales características tenían más peso a la hora de diferenciar entre voces *bonafide* y voces *deepfake.*

E. Ajustar hiperparámetros

El ajuste de los hiperparámetros fue una parte fundamental del estudio. Este proceso fue esencial para optimizar el rendimiento de los modelos LSTM y de *machine learning*, garantizando que funcionaran de manera eficiente en la tarea de detección de voces generadas por IA.

Para el modelo LSTM, se evaluaron diferentes tasas de aprendizaje, siempre utilizando el optimizador ADAM. Esto permitió obtener un balance entre tiempo de entrenamiento, y rendimiento del modelo. Además, se implementó un "*Early Stop*" que utilizó un umbral para detener el entrenamiento si no se observaban mejoras significativas en el conjunto de validación después de 3 épocas. Esta técnica ayudó a ahorrar recursos computacionales y de tiempo además de evitar el sobreajuste del modelo.

Para el modelo SVM, el ajuste de hiperparámetros se enfocó en ajustar hiperparámetros como el tipo de *kernel* (lineal, radial o polinomial) y el parámetro de regularización “C”. El tipo de *kernel* fue crucial, ya que definió cómo el modelo mostraba las características en un espacio de mayor dimensionalidad para realizar la clasificación. Se probaron diferentes *kernels* para identificar cuál proporcionaba el mejor rendimiento. El parámetro de regularización controló la flexibilidad del margen de decisión; valores más altos permitían un margen más estrecho, mientras que valores más bajos permitían un margen más amplio.

Por otro lado, para los modelos creados por el Autogluon, esta librería ofrece optimización de hiperparámetros por defecto por lo cual no fue necesario probar con diferentes opciones para cada modelo. No obstante, es importante mencionar que para lograr esto, se dejó correr al modelo de Autogluon por aproximadamente una hora completa para que pudiera encontrar los mejores hiperparámetros para cada modelo.

Finalmente, una vez identificados los mejores hiperparámetros para cada método, se realizó un entrenamiento final utilizando estos parámetros optimizados. Los modelos ajustados se evaluaron en el conjunto de prueba para confirmar que los ajustes mejoraron significativamente el rendimiento, y los resultados finales se compararon para determinar

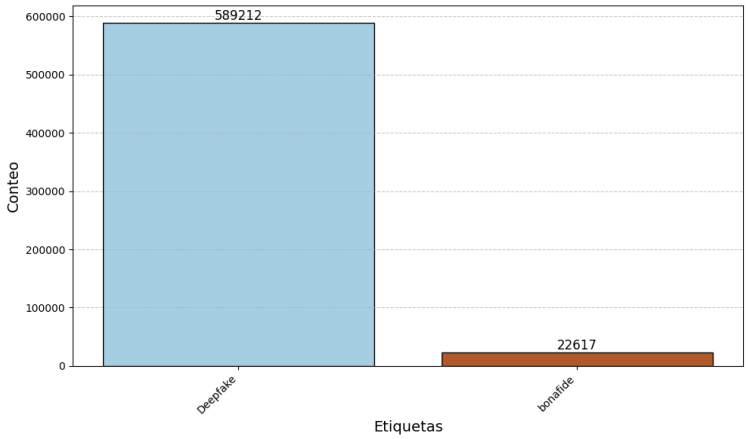
qué modelo ofrecía la mejor combinación de precisión y eficiencia en la detección de voces generadas por IA.

VI. Resultados:

A. Análisis Exploratorio:

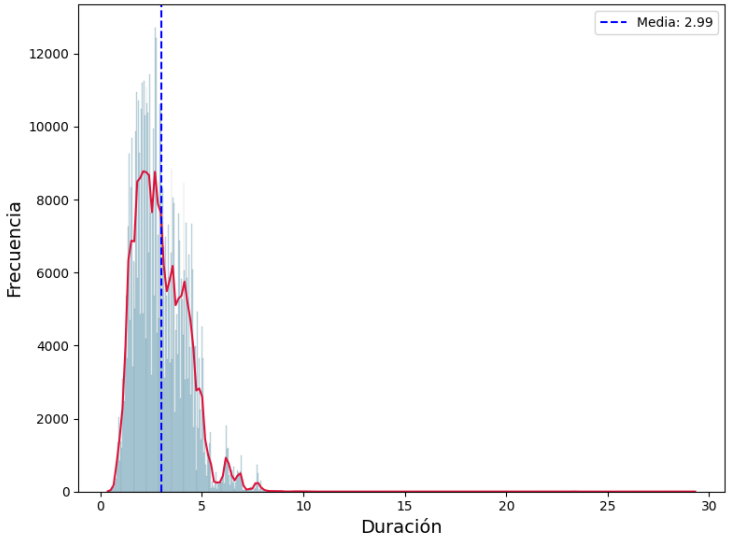
La Figura 3 muestra la distribución del conjunto de datos en dos categorías: *deepfake* y *bonafide*. Como se observa, la mayoría de los datos pertenecen a la categoría de *deepfake*, con un total de 589,212 ejemplos, en comparación con los 22,617 ejemplos de la categoría *bonafide*.

Figura 3: Cantidad de datos por cada etiqueta (*deepfake* o bonafide)



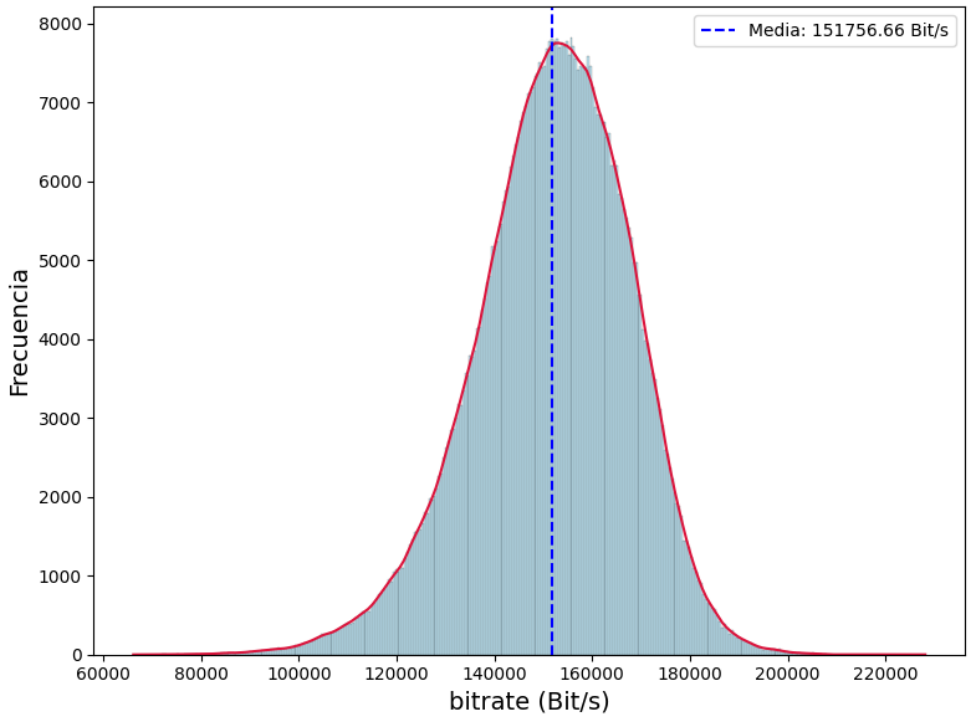
La Figura 4 presenta la distribución de la duración de los audios en el conjunto de datos total antes de ser preprocesado. La mayoría de los audios tienen una duración inferior a 5 segundos, con una media de 2.99 segundos (indicada por la línea punteada azul).

Figura 4: Distribución de la duración de los audios



La Figura 5 muestra la distribución del *bitrate* (tasa de bits) de los archivos de audio en el conjunto de datos. El *bitrate* representa la cantidad de información que se procesa por segundo en un archivo de audio y se mide en bits por segundo (Bit/s) En este gráfico, se observa una distribución aproximadamente normal, con una media de 151,756.66 Bit/s (indicada por la línea punteada azul). La densidad de la curva (línea roja) muestra un pico alrededor de la media, indicando que la mayoría de los archivos de audio tienen una calidad similar en términos de *bitrate*.

Figura 5: Distribución del *Bitrate* (Cantidad de información en el audio)



El Cuadro 2 muestra la media de diferentes características de audio para las dos etiquetas del conjunto de datos, deepfake y bonafide. Se tienen características como, Zero Cross-Rate (ZCR), el Root Mean Square Energy (RMSE), características, características relacionadas al espectro de audio, las Características Cromáticas y los Coeficientes Cepstrales en la Frecuencia Mel (MFCCs) del 1 al 10. Se pueden notar diferencias en las medias entre las dos categorías, tomando el MFCC 1, se tiene una diferencia de casi 60 unidades. Por otro lado, se puede observar el Spectral Bandwith, el cual presenta una diferencia de casi 150 unidades entre categorías. Cabe resaltar que por lo general todas las características se mantienen en los negativos o positivos para cada categoría, exceptuando dos, MFCC 8 y MFCC 5, los cuales se encuentran entre los números negativos para la categoría Deepfake mientras que para la Bonafide se encuentra dentro de los positivos.

Cuadro 2: Media de cada característica por cada etiqueta del conjunto de datos sin filtro aplicado

Característica	*Deepfake*	*Bonafide*
ZCR	0.103894	0.087593
RMSE	0.115821	0.086005
Spectral Centroid	1240.050837	1262.641777
Spectral Bandwith	1086.071473	1234.275227
Spectral Rollof	2298.883281	2534.975936
Spectral Contrast	30.701906	28.662093
Chroma Features	0.398647	0.470517
MFCC 1	-260.606659	-326.489105
MFCC 2	123.984085	99.319687
MFCC 3	-26.845573	-16.354847
MFCC 4	39.074932	34.219452
MFCC 5	-3.618037	0.009340
MFCC 6	12.030872	10.750428
MFCC 7	-7.794425	-2.746076
MFCC 8	-2.272178	0.794871
MFCC 9	-10.489719	-7.9110810
MFCC 10	-12.195688	-5.033755

B. Modelo SVM

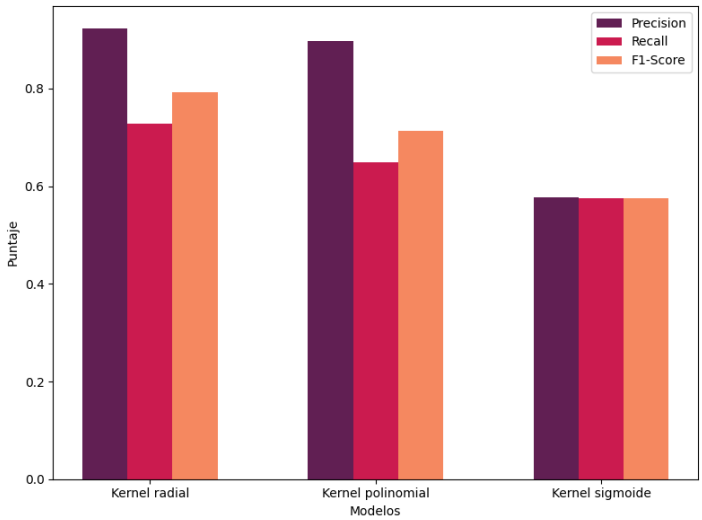
El Cuadro 3 presenta las métricas de desempeño obtenidas al evaluar el modelo *Support Vector Machine* (SVM) utilizando tres tipos de kernels: Radial, Polinomial, y Sigmoide, sobre el conjunto de datos de prueba sin ningún filtro estadístico aplicado. Se observa que en todas las métricas para medir las predicciones del modelo el SVM con *kernel* radial fue el que obtuvo el puntaje más alto. Asimismo, al tomar en cuenta el tiempo de predicción de 0.010590 segundos y el tiempo de entrenamiento de 8.79 minutos, este modelo fue el que presentó predicciones y tiempo de entrenamiento más rápidos.

Cuadro 3: Métricas de los diferentes *kernels* del modelo SVM para el conjunto de datos de prueba sin filtros.

Kernel	***Accuarcy***	***Macro-avg Precision***	***Macro-avg Recall***	***Macro-avg F1-Score***	***Pred. Time (s)***	***Train time (min)***
Radial	0.9742	0.9229	0.7275	0.7933	0.010590	8.79
Polinomial	0.9678	0.8981	0.6494	0.7127	0.005295	11.84
Sigmoide	0.9317	0.5766	0.5758	0.5762	0.009698	10.57

La Figura 6 muestra la comparación de las métricas de desempeño de los modelos SVM con tres diferentes *kernels*: radial, polinomial y sigmoide. Las métricas consideradas son precision, recall y F1-Score. Se puede observar que el modelo con *kernel* sigmoide presenta las métricas más agrupadas con un puntaje de alrededor de 0.57 para cada una, sin embargo, son las más bajas. Por otro lado, los otros 2 *kernels* muestran métricas más variadas, siempre teniendo precision como la más alta y recall como la más baja.

Figura 6: Comparación de las métricas de los modelos SVM con diferente Kernel



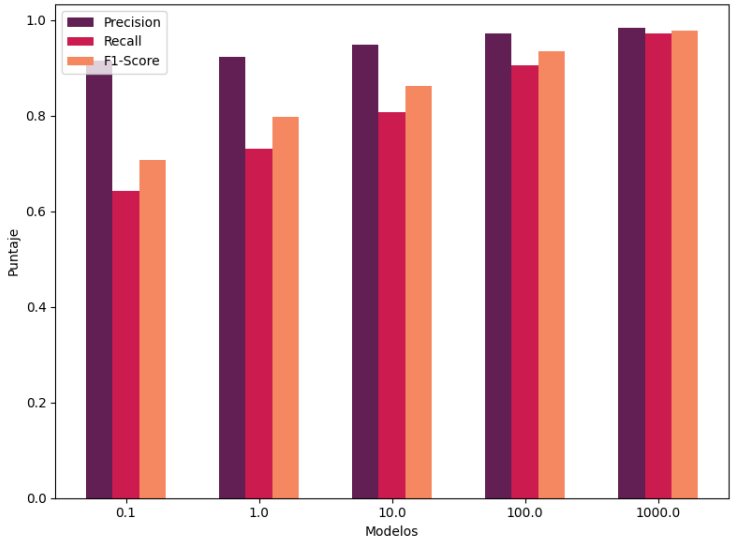
El Cuadro 4 presenta las métricas de desempeño del modelo SVM con *kernel* radial para diferentes valores del coeficiente de penalización "C", utilizando el conjunto de datos de prueba. Se pude observar que, para los distintos coeficientes, la métrica de *accuracy* se mantiene en un rango parecido, sin embargo, para las otras métricas empieza a variar más. No obstante, el modelo con un coeficiente de 1000 no solo presenta las métricas más altas sino también tiene el tiempo de predicción más rápido. Pese a tener buen desempeño, este modelo también presenta el tiempo de entrenamiento más alto de 105 minutos, una diferencia de 97 minutos del modelo con coeficiente de 0.1, el cual presenta el tiempo más bajo.

Cuadro 4: Métricas de los diferentes coeficientes "C" del modelo SVM con *kernel* radial, para el conjunto de datos de prueba sin filtros

Coeficiente C	***Accuarcy***	***Macro-avg Precision***	***Macro-avg Recall***	***Macro-avg F1-Score***	***Pred. Time (s)***	***Train time (min)***
0.1	0.9679	0.9152	0.6421	0.7065	0.001176	8.80
1	0.9744	0.9233	0.7311	0.7965	0.000947	8.79
10	0.9813	0.9485	0.8065	0.8630	0.000813	15.31
100	0.9903	0.9717	0.9047	0.9354	0.000664	42.18
1000	0.9964	0.9834	0.9716	0.9774	0.000507	105.30

La Figura 7 presenta la comparación de las métricas de desempeño del modelo SVM con *kernel* radial para diferentes valores del coeficiente de penalización C. Las métricas representadas son *precision*, *recall* y *F1-Score*. Se puede ver una tendencia a medida que se incrementa el valor C, también se incrementan todas las métricas de desempeño de los modelos. Junto a esa misma tendencia se puede observar que diferencia entre cada una de las métricas va disminuyendo. Asimismo, para cada uno de los modelos de observa que se mantiene un orden en cuanto a puntaje, siendo *precision* el más alto y *recall* el más bajo, mientras que *f1-score* se mantiene entre estos dos.

Figura 7: Comparación de métricas del modelo SVM con kernel radial y distintos valores de coeficiente C



C. Otros modelos de machine learning

El Cuadro 5 muestra las métricas de desempeño de tres modelos base: Selección del más frecuente, Aleatorio, y Regresión Logística. Las métricas con las que se midió el desempeño de los modelos fueron las mismas que se usaron para modelos anteriores. Las métricas de estos modelos serán utilizadas para fijar una base para comparar los resultados de otros modelos. Como se puede observar tanto Selección del más frecuente (predice la categoría *deepfake* siempre) como Regresión logística tienen una métrica de *accuracy* alta, sin embargo, decaen en las demás métricas. Asimismo, se puede observar que el modelo que escoge aleatoriamente presenta métricas cercanas al 0.5. Por otro lado, la regresión Logística presenta el tiempo de entrenamiento y de predicción más altos con 1 segundo y 0.003 segundos respectivamente. Mientras que los otros dos modelos presentan tiempos de entrenamiento y de predicción similares siempre más bajos que el modelo de Regresión Logística.

Cuadro 5: Métricas de modelos base

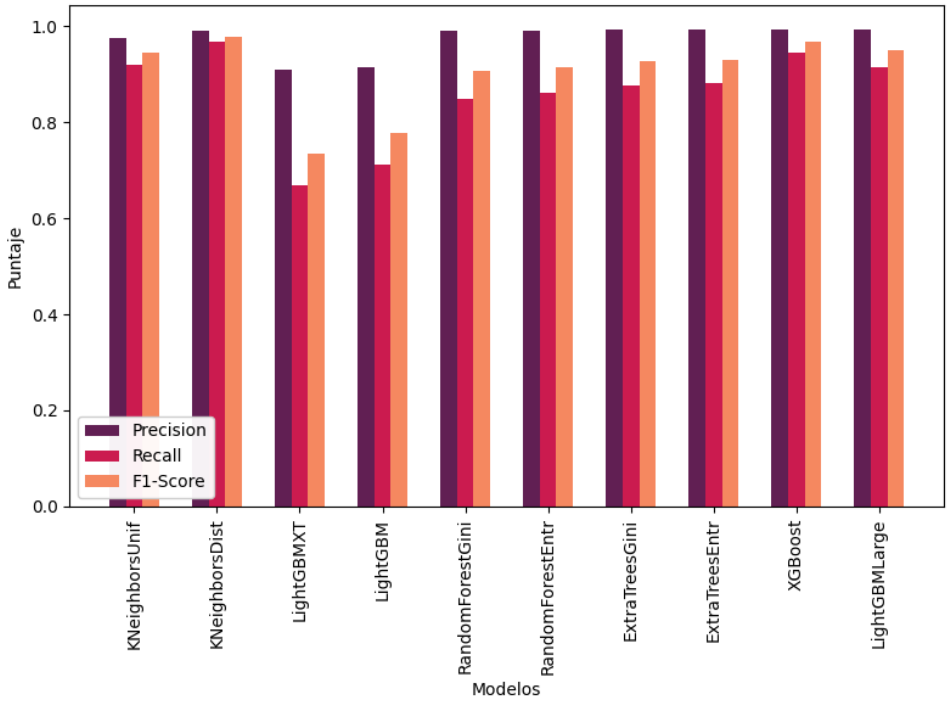
Modelo	*Accuarcy*	*Macro-avg Precision*	*Macro-avg Recall*	*Macro-avg F1-Score*	*Pred. Time (s)*	*Train time (s)*
Selección del más frecuente	0.9577	0.4789	0.5000	0.4892	0.000317	0.0127
Aleatorio	0.4989	0.5002	0.5014	0.3671	0.000285	0.0176
Regresión logística	0.9570	0.7165	0.5645	0.5942	0.003537	1.0626

El Cuadro 6 muestra las métricas de desempeño de varios modelos de *machine learning* generados automáticamente utilizando la librería de Autogluon en Python. Estos modelos incluyen variantes de *KNeighbors*, *LightGBM*, *RandomForest*, *ExtraTrees*, y *XGBoost*. Por lo general se puede observar que todos los modelos tienen un desempeño similar para el set de datos de prueba. Sin embargo, los modelos de *LightGBMXT* y *LightGBM* presentan los resultados más bajos en las métricas de *precision, recall* y de *F1-Score*. Esto es aún más aparente cuando se observa la Figura 8 la cual muestra estas para todos los modelos vistos y son aparentes que los dos modelos mencionados anteriormente son los más bajos. Asimismo, el tiempo de predicción se mantiene parecido a lo largo de los modelos exceptuando las variaciones de modelos de *ExtraTrees* y de *RandomForest*, las cuales presentan tiempos de predicción desde 0.85 hasta 1.76 segundos. Incluso, al observar el tiempo de entrenamiento, estos modelos junto al *XGboost* son los que presentan el tiempo más alto, desde 72 segundos hasta 670 segundos para las variaciones del *RandomForest*. No obstante, hay un modelo que destaca en todas las métricas de desempeño, *KneighborsDist* el cual no solo tiene las métricas de predicción más altas sino también el tiempo de entrenamiento más bajo y uno de los tiempos más bajos en predicción.

Cuadro 6: Métricas de los diferentes modelos de machine Learning creados con Autogluon

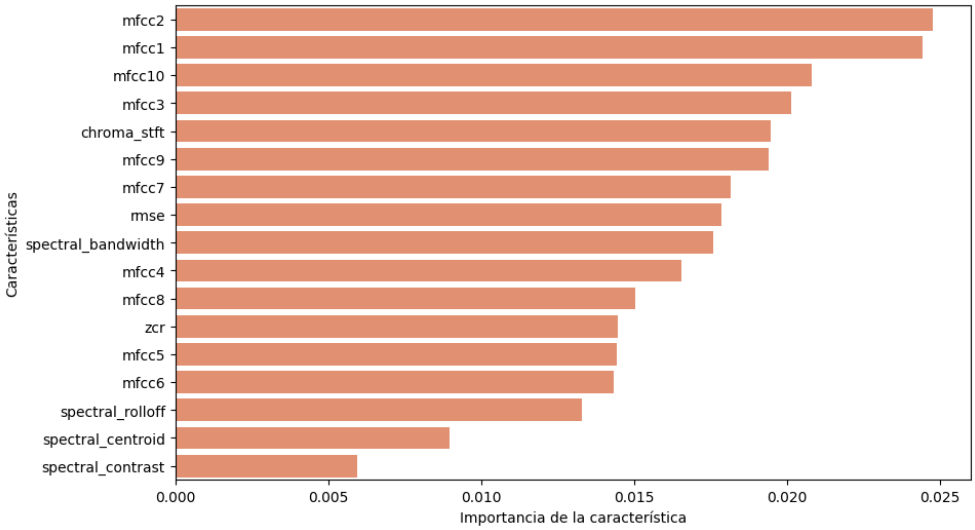
Modelo	*Accuarcy*	*Macro-avg Precision*	*Macro-avg Recall*	*Macro-avg F1-Score*	*Pred. Time (s)*	*Train time (s)*
KneigborsUnif	0.9916	0.9747	0.9189	0.9449	0.064006	0.4556
KNeighborsDist	0.9967	0.9919	0.9674	0.9793	0.066382	0.4319
LightGBMXT	0.9695	0.9102	0.6679	0.7341	0.015085	10.1921
LigthGBM	0.9727	0.9156	0.7117	0.7776	0.018300	11.8712
RandomForestGini	0.9870	0.9916	0.8478	0.9063	0.852141	658.5111
RandomForestEntr	0.9881	0.9915	0.8613	0.9156	0.725383	674.7434
ExtraTreesGini	0.9896	0.9946	0.8769	0.9271	1.763671	94.8682
ExtraTreesEntr	0.9899	0.9946	0.8811	0.9298	1.610336	127.5120
XGBoost	0.9951	0.9934	0.9455	0.9681	0.147202	72.4607
LightGBMLarge	0.9926	0.9935	0.9147	0.9504	0.181014	1.1360

Figura 8: Comparación de los modelos de *machine learning* obtenidos con el Autogluon



La Figura 9 muestra la importancia de las características utilizadas en los modelos de clasificación para el conjunto de datos de prueba. En el gráfico, se observa la contribución relativa de cada característica a la predicción del modelo. En los primeros tres lugares se tienen los Coeficientes Cepstrales en la Frecuencia Mel, específicamente el 2, 1 y 10; todos con un valor mayor a 0.020. Por otro lado, entre las características que menos aportan se encuentran algunas de las características relacionados con el espectro de audio, teniendo a *spectral centroid* y *spectral contrast* con un valor menor a 0.010.

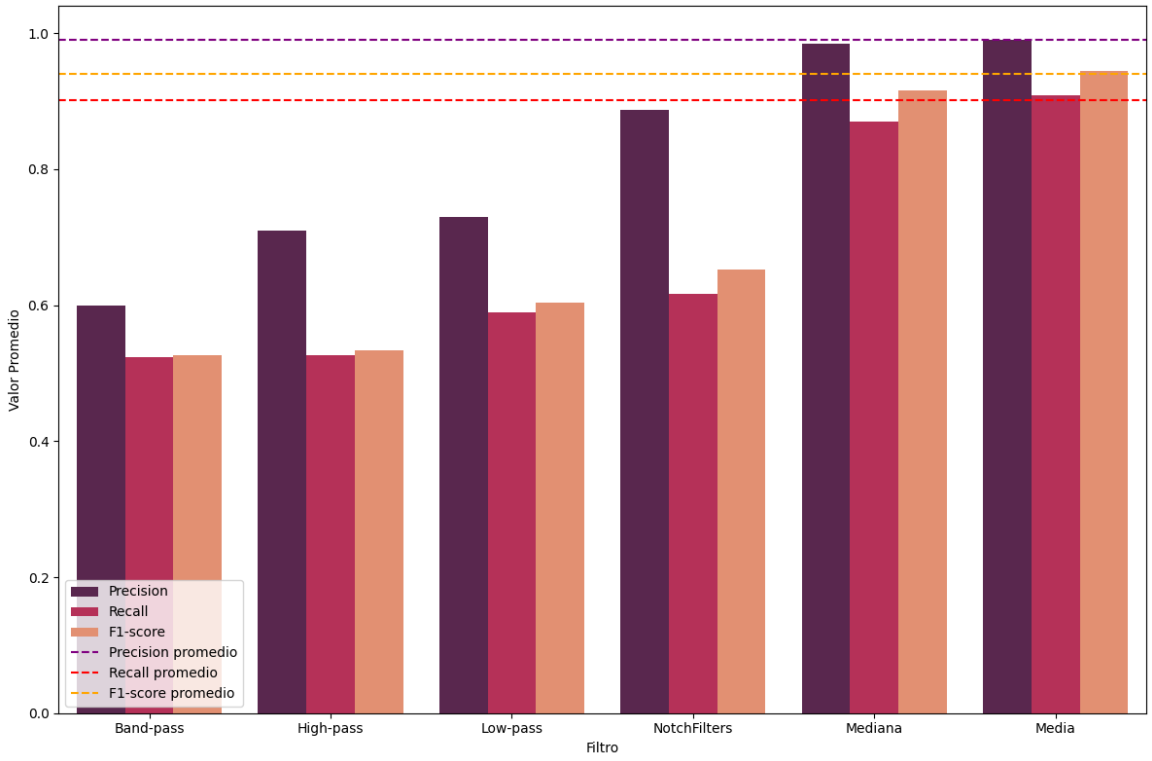
Figura 9: Importancia de las características para el conjunto de datos de prueba



La Figura 10 muestra una comparación de los rendimientos que obtenidos para los modelos generados por el Autogluon utilizando conjuntos de prueba con distintos filtros aplicados. Es importante señalar que los gráficos de barras representan tan solo el rendimiento promedio de las variaciones de cada filtro. Para este caso, se trabajó con distintas variaciones para cada filtro, específicamente 4 variaciones. Para los filtros de media y mediana se utilizaron diferentes ventanas de desplazamiento para calcular el valor de media y mediana locales. Los tamaños de ventana con los que se trabajaron fueron de 3, 5, 7 y 11 datos. Por otro lado, para los otros filtros se trabajó también con 4 variaciones, sin embargo, se probaron diferentes umbrales obtenidos de las estadísticas de las características de las dos categorías. Por ejemplo, para el filtro *Band-pass* se utilizaron 2 variaciones usando estadísticas únicamente de la categoría *deepfake* y otras 2 solo con la categoría *bonafide*.

El gráfico muestra que, por lo general, los filtros que dejan pasar solo cierto tipo de datos como: *Band-pass, High-pass, Low-pass* y *Notch*, tienen un rendimiento bajo en comparación de los otros dos. De los cuatro filtros mencionados anteriormente, el filtro *Notch* fue el que mejor rendimiento tuvo de ese grupo. Por otro lado, si se observan los filtros de media y mediana, se puede notar que tienen rendimientos casi iguales, con sus mayores diferencias visibles siendo las métricas de *recall* y *F1-score*. Asimismo, todos los filtros se encuentran igual o por debajo de las métricas promedio que se obtuvieron con el conjunto de datos de prueba sin filtros (denotado por las líneas punteadas), demostrando que no hubo una mejora en el rendimiento al aplicar los filtros.

Figura 10: Comparación de los rendimientos promedio de los filtros estadísticos aplicados.



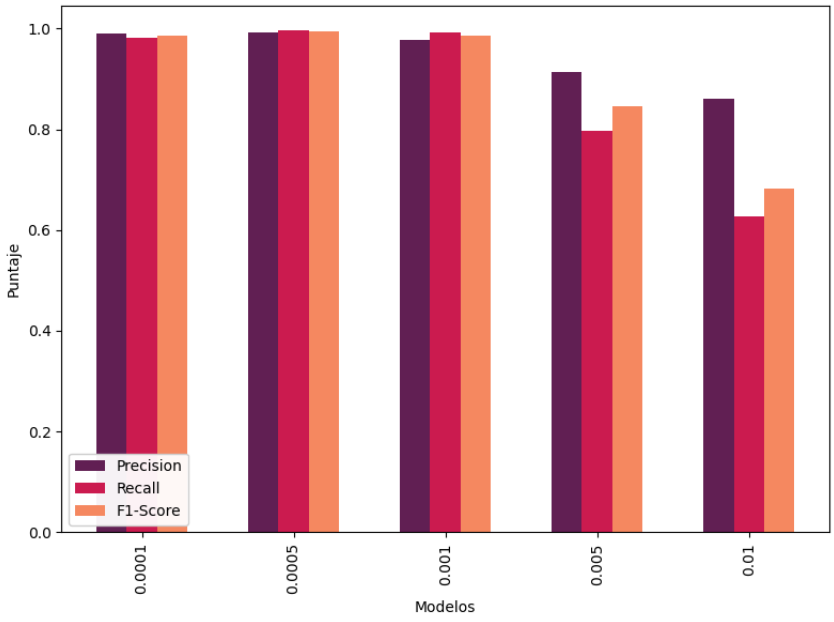
D. Modelo utilizando red neuronal LSTM

El Cuadro 7 presenta las métricas de rendimiento del modelo Long Short-Term Memory (LSTM) utilizando el optimizador ADAM con distintos pasos de aprendizaje. En esta tabla se puede observar que los modelos que utilizaron pasos de aprendizaje de 0.0001 hasta 0.001 tienen puntajes similares en todas las métricas, incluyendo tiempo de predicción y tiempo de entrenamiento. Esto es aún más evidente cuando se observa la Figura 11 en el cual se puede ver una grada en el rendimiento entre el modelo con pazo de aprendizaje de 0.001 y el modelo con 0.005. No obstante, pese a que los modelos que presentan métricas más bajas tienen a su vez un tiempo de entrenamiento más bajo ya que el Early Stopping que se le agregó a los modelos detuvo su entrenamiento. Mientras que para los otros si se entrenaron por todas las épocas establecidas.

Cuadro 7: Métricas del modelo LSTM con optimizador ADAM

Paso de aprendizaje	***Accuarcy***	***Precision***	***Recall***	***F1***	***Pred. Time (s)***	***Train time (min)***
0.0001	0.9977	0.9904	0.9816	0.9859	0.052977	19.4874
0.0005	0.9990	0.9928	0.9954	0.9941	0.054845	19.5644
0.001	0.9975	0.9781	0.9920	0.9850	0.055967	19.6881
0.005	0.9785	0.8142	0.7975	0.8453	0.056977	10.7763
0.01	0.9650	0.8612	0.6261	0.6821	0.057450	6.9026

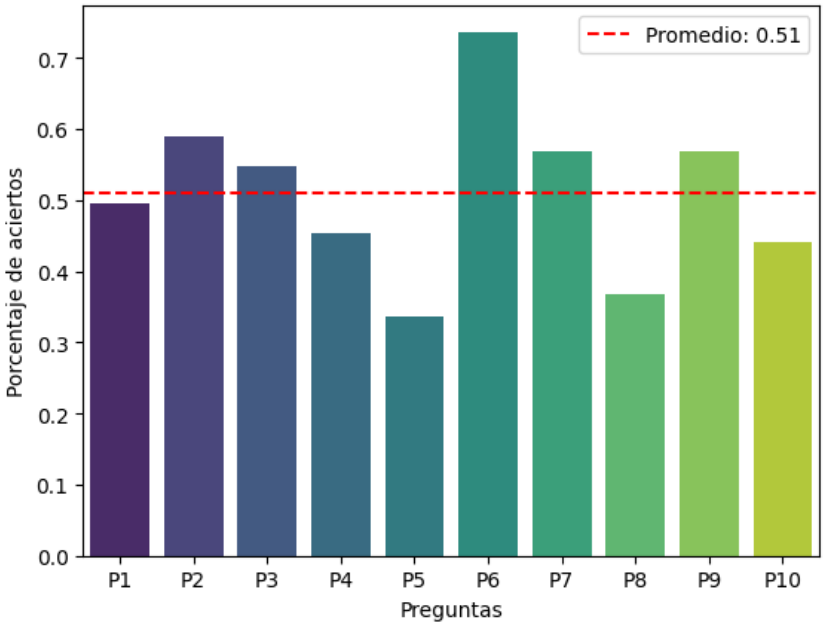
Figura 11: Comparación del modelo LSTM con diferentes pasos de aprendizaje



E. Reconocimiento humano

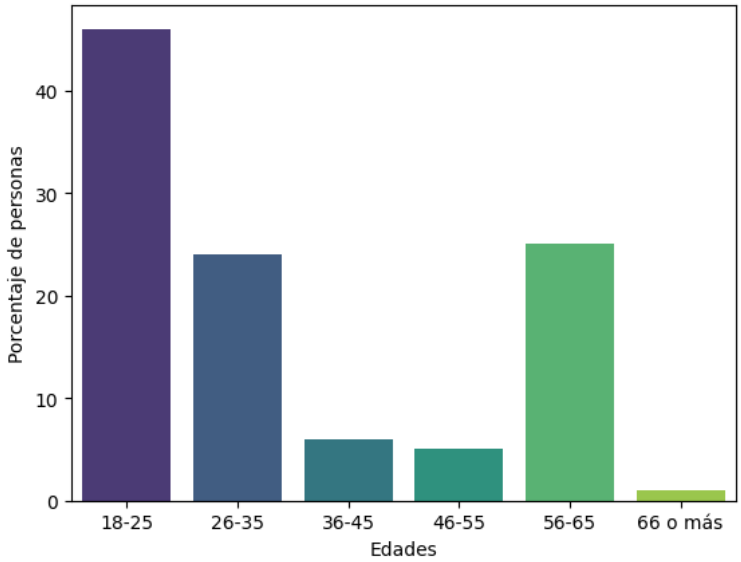
La Figura 12 muestra el porcentaje de aciertos por cada una de las preguntas del examen de reconocimiento humano, en el que los participantes tuvieron que diferenciar si las voces eran reales o generadas por IA. La línea roja indica el promedio general de aciertos, la cual se sitúa en 0.51 aciertos, es decir 5 de cada 10 preguntas fueron aciertos. Como se muestra en el gráfico, la pregunta número 6 fue la que más porcentaje de aciertos tuvo mientras que la pregunta 5 fue la que menos tuvo con aproximadamente el 30% de los participantes acertando. Se observa que 5 de las 10 preguntas se encuentran arriba del promedio en aciertos las cuales son la pregunta 2, 3, 6, 7 y 9 las cuales todas son voces generadas por inteligencia artificial. Por otro lado, todas aquellas preguntas debajo de la línea de promedio pertenecen a la categoría de voces reales.

Figura 12: Porcentaje de aciertos por cada pregunta del examen de reconocimiento humano



La Figura 13 muestra el porcentaje de participantes pertenecientes a cada uno de los grupos de edades. Como se puede observar, el grupo de personas de 18-25 años es el que presenta más personas abarcando casi un 45% de los participantes totales, siguiéndole el grupo de 26 a 35 años y el de 56 a 45 años. Por otro lado, el grupo de 66 o más corresponde a aproximadamente el 1% de los participantes, siendo por mucho el grupo con menos participantes para este examen.

Figura 13: Porcentaje de personas por cada edad



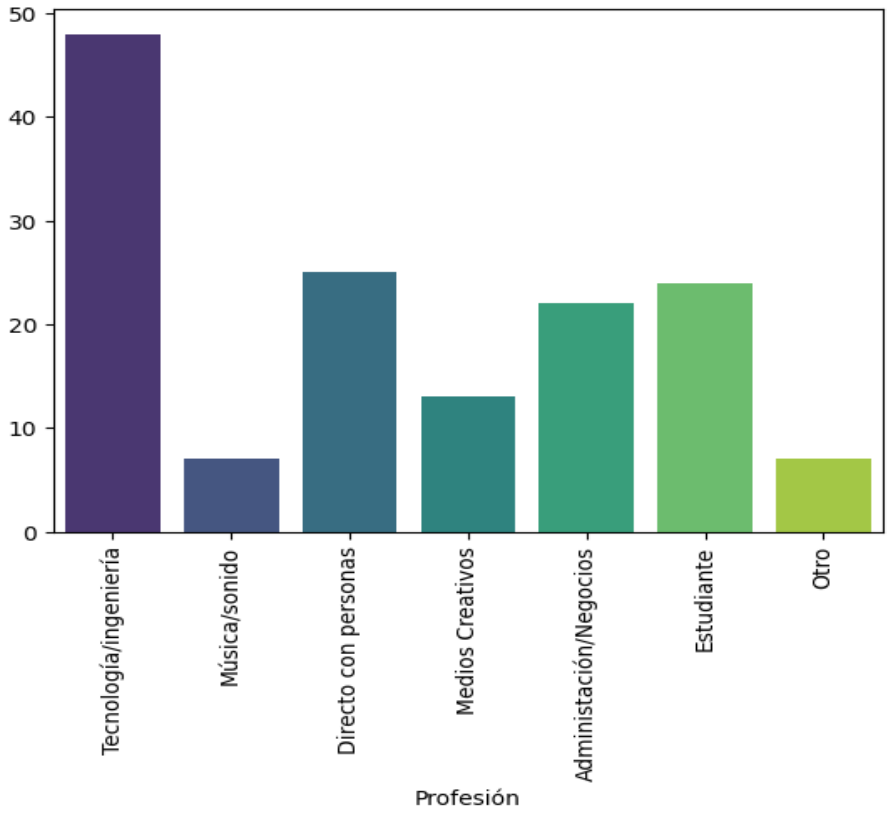
El Cuadro 8 presenta las métricas de los grupos de edades al contestar el examen. Como se puede observar, el grupo de 66 o más personas presenta las métricas más altas por mucho teniendo 0.8, aunque, como se mencionó anteriormente solo representa al 1% de los participantes. Por otro lado, el segundo grupo más alto fue el de las personas con edades entre 18 a 25 años, teniendo aproximadamente un puntaje de 0.61 para todas las métricas. En cuanto a los otros grupos, sus métricas se mantienen en un rango similar, de entre 0.49 para el grupo de 56 a 65 hasta 0.54 para el grupo de 46 a 55.

Cuadro 8: Métricas del examen de reconocimiento humano agrupado por edad

Edad de las personas	***Accuarcy***	***Precision***	***Recall***	***F1***
18-25	0.6125	0.6133	0.6125	0.6118
26-35	0.5200	0.5200	0.5200	0.5198
36-45	0.5167	0.5171	0.5167	0.5133
46-55	0.5400	0.5401	0.5400	0.5398
56-65	0.4913	0.4911	0.4913	0.4885
66 o más	0.8000	0.8000	0.8000	0.8000

La Figura 14 muestra la distribución de los participantes del estudio según su profesión. Cabe mencionar que los participantes podían elegir más de una profesión dependiendo de lo que pensaban que más describiría sus actividades. El grupo que obtuvo la mayor cantidad de personas fue el de tecnología/ingeniería con aproximadamente 50 personas. Por otro lado, los grupos de: Trabajo directo con personas, Administración/negocios y estudiantes tuvieron cantidad de participantes parecidas, con un poco más de 20 por cada uno. Asimismo, los grupos con menos participantes fueron Música/sonido y otro, ambos con menos de 10 personas por cada grupo.

Figura 14: Cantidad de personas por cada profesión



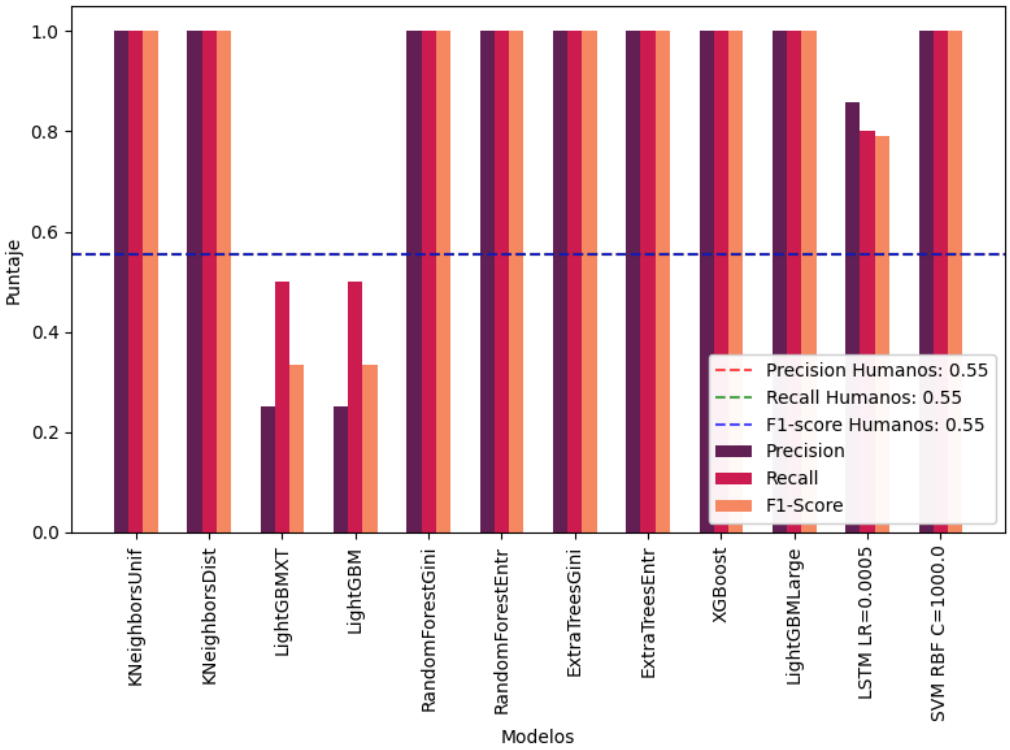
El Cuadro 9 muestra las métricas del examen de reconocimiento humano por cada grupo de profesión que se podía seleccionar. Es importante aclarar que los resultados fueron tomados de todas aquellas personas que eligieron un grupo durante el examen, es decir que, por ejemplo, los resultados de música/sonido incluyan personas que también seleccionaron otra categoría y viceversa. Por lo tanto, se puede observar que dentro de los grupos que tuvieron mejor rendimiento en el examen se encuentra el grupo de tecnología/ingeniería y también el grupo de estudiantes. Por otro lado, de los grupos que menos rendimiento tuvieron fueron aquellos que relacionaban su profesión con algún medio creativo, teniendo el puntaje más bajo por mucho. Por lo general, se puede observar que las métricas para cada uno de los grupos se mantienen en el mismo rango, sino que son exactamente las mismas con tan solo unas variaciones menores a una milésima.

Cuadro 9: Métricas del examen de reconocimiento humano agrupado por profesión

Categoría de profesión	*Accuarcy*	*Precision*	*Recall*	*F1-score*
Tecnología/ingeniería	0.5917	0.5918	0.5917	0.5915
Música/sonido	0.5143	0.5143	0.5143	0.5139
Directo con personas	0.5520	0.5521	0.5520	0.5519
Medios creativos	0.4462	0.4461	0.4462	0.4460
Administración/negocios	0.5455	0.5455	0.5455	0.5455
Estudiante	0.5833	0.5835	0.5833	0.5831
Otro	0.5000	0.5000	0.5000	0.5000

La Figura 15 presenta una comparación entre los resultados de la prueba de reconocimiento humano y el desempeño de los modelos para identificar los audios utilizados en dicho examen. Como se puede observar la mayoría de los modelos a excepción de *LightGBMXT* y *LightBGM*, tienen mejores métricas que los participantes del examen los cuales obtuvieron un puntaje de 0.55 en todas las métricas, denotado por la línea punteada. Por otro lado, de los modelos que obtuvieron mejor puntaje que las personas solo uno no obtuvo un puntaje perfecto, el modelo *Long Short-Term Memory* (LSTM), el cual obtuvo un puntaje de 0.8 para *precision* mientras que para *recall* y *F1-Score* obtuvo un puntaje menor. Cabe resaltar que para esta comparación solamente se utilizaron los modelos que obtuvieron mejores métricas en sus pruebas mencionadas anteriormente. Es decir que se utilizaron los modelos del Autogluon junto con el modelo LSTM con pazo de aprendizaje de 0.0005 y el SVM con *kernel* radial y con un coeficiente de penalización de 1000.0.

Figura 15: Comparación de métricas de modelos con los datos del examen de reconocimiento humano



VII. Discusión

En este estudio se evaluaron diferentes métodos para identificar voces generadas por inteligencia artificial, los cuales incluyen modelos de machine learning con filtros estadísticos aplicados, redes neuronales LSTM y el reconocimiento humano. Los resultados demostraron que el modelo de *KneighborsDist* proporcionado por la librería Autogluon de Python fue el más eficiente, destacando tanto en métricas de predicción como en términos de consumo de recursos. Por otro lado, el modelo LSTM presentó altos puntajes en métricas de precisión, pero presentó indicios de sobreajuste al trabajar con datos no vistos y además, requirió la mayor potencia computacional al entrenarse con una GPU. Por último, el reconocimiento humano tuvo el rendimiento más bajo entre los métodos estudiados, sin embargo, ofrece información valiosa sobre la habilidad de las personas en la detección de voces *deepfake* sin asistencia tecnológica. Estos resultados sugieren que para este tipo de problema los modelos de machine learning sin ningún filtro aplicado representan la mejor solución con una alta capacidad de generalización y menor consumo de recursos.

Al realizar el análisis exploratorio se pudo encontrar que los datos de las dos clases que se estaban observando (*deepfake* y *bonafide)* se encontraban desbalanceadas tal y como se puede observar en la Figura 3. Incluso después del preprocesamiento de los datos, se trabajó con aproximadamente la mitad de los audios originales, sin embargo, la proporción de las categorías permaneció igual. Además, observando la distribución de la duración de audios se pudo saber que existía algo parecido a una distribución normal una vez se removían los datos atípicos, tal y como se puede observar en la Figura 4. Esto da a entender que la mayoría de los audios tienen una duración de entre uno a cinco 5 segundos. Esto es útil de saber debido a que nos permite establecer un parámetro a cerca de que tanta información de audio se necesita para que un modelo pueda hacer una predicción correcta. Por otro lado, la calidad de audio en cuanto a *bitrate* es parecida para la mayoría de los audios, tal y como se muestra en la Figura 5. De esta manera se puede determinar que las diferencias observadas en el rendimiento de los modelos no se deben a una variación en la calidad de audio, sino que en todo caso reflejan las diferencias entre las dos categorías del conjunto de datos.

Por otro lado, se puede obtener las características que más están afectando a la diferencia del modelo. Observando el Cuadro 2, se puede estimar la diferencia entre la media de cada una de las características para cualquiera de las etiquetas observadas en esta investigación. Estas diferencias en las medias sugieren que, si existe una diferencia en los datos de cada característica entre etiquetas, lo cual determinaría que los modelos de *machine learning* y redes neuronales podrán clasificar correctamente los audios.

Continuando con los modelos de *machine learning* como *Support Vector Machines*, se pueden obtener varios datos valiosos en cuanto al tiempo de entrenamiento, tiempo de decisión y bien, métricas del modelo en general. Como se puede observar, el Cuadro 3, se compararon diferentes *kernels* para saber cuál era el que mejor se ajustaba a los datos proporcionados. Dentro de los resultados obtenidos se determinó que el kernel radial, tuvo un rendimiento más alto, con puntaje de *accuracy* del 0.9742 y un puntaje de *precision* del 0.9229. E incluso, se logró un tiempo de 8 minutos siendo entrenado bajo un ambiente de Google Colab que utiliza CPU. Sin embargo, las métricas de *recall* y *F1-Score* se encuentran más bajas, con un puntaje de 0.7275 y 0.7933 respectivamente.

La diferencia que se observa entre estas métricas es debido a que el modelo logra reconocer de manera correcta los audios *deepfake,* sin embargo, algunos audios reales también los categoriza como *deepfake* erróneamente. En un conjunto de datos tan desbalanceado como el trabajado es mejor utilizar las métricas de *recall* y *F1-Score* pues le agregan peso a aquellos fallos de la categoría minoritaria. Incluso, para verificar mejor el rendimiento del modelo se puede utilizar *Macro-Average*, esta métrica establece el promedio de cada métrica individual (*precision*, recall y *F1-Score*) calculada para cada clase, sin ponderar por la cantidad de muestras en cada una. Esto significa que el *Macro-Average* les da el mismo peso a ambas clases, independientemente de si están desbalanceadas.

Utilizando el modelo SVM con *kernel* radial como base, se pudo ajustar el hiperparámetro "C" o también conocido como parámetro de regularización. Este hiperparámetro controla la flexibilidad del modelo o bien la rigidez que se tiene al elegir un hiperplano que pueda dividir los datos. Mientras más sube el coeficiente, más rígido se es con las predicciones, es decir que se trata de encontrar un hiperplano que incluya menos errores de predicción, sin embargo, esto puede llevar al sobreajuste. Esto es evidente, si se observa la Figura 7, en donde a medida que aumenta el coeficiente C, las métricas también aumentan. Sin embargo, al observar el Cuadro 4, donde se puede observar que el tiempo de entrenamiento a medida que sube el valor de C también aumenta. Por lo tanto, el modelo con un C = 1000.0, presenta el mayor puntaje en todas las métricas y también el tiempo de entrenamiento más alto. No obstante, como se mencionó anteriormente, el utilizar un C tan grande, puede tender a generar un sobreajuste a los datos de entrenamiento y de esta manera no se podría generalizar bien. Por lo tanto, utilizar un modelo pese a que el modelo presente resultados prometedores incluso con el conjunto de datos de prueba, puede que para otro conjunto de datos no se tengan estos resultados. Por lo tanto, podría ser más útil utilizar un C más bajo para balancear la generalización y las métricas de predicción como podría ser el C=100.0 que también tiene métricas altas, pero puede que evite el sobreajuste y además su tiempo de entrenamiento por lo general será más bajo.

Para lograr entender el rendimiento de los modelos, se realizaron modelos base los cuales se "entrenaron" y probaron con el mismo conjunto de datos que otros modelos. Hacer esto presenta una métrica con la cual comparar el rendimiento de modelos, e incluso entender mejor los resultados obtenidos. Se realizaron 3 modelos, el primero que su predicción siempre era la categoría con más datos, en este caso *deepfake*. El segundo modelo que sus predicciones se realizaban aleatoriamente y por último un modelo de regresión logística simple. Al observar el Cuadro 5, el modelo que tuvo puntaje más alto en *accuracy* fue el modelo de selección del más frecuente. Esto evidencia por qué esta métrica

no es útil al monitorear el rendimiento de los modelos ya que se pueden obtener valores altos sin ningún tipo de aprendizaje. Asimismo, si se observa el tiempo de entrenamiento y de predicción, se puede obtener un tiempo mínimo que le debe tomar a un modelo hacer una predicción o bien entrenarse. Esto se debe a que no se realiza ninguna operación para obtener el resultado y por lo tanto el tiempo de predicción simplemente es el tiempo que le toma al programa retornar la categoría mayoritaria.

Al observar los otros modelos que se probaron a través del Autogluon, los resultados mostrados en el Cuadro 6 muestran que los dos mejores modelos en cuanto a rendimiento fueron *KneighborsDist* y *XGBoost*. Estos dos modelos, no solo presentan las mejores métricas, sino que *KneighborsDist* también presenta el tiempo de entrenamiento más bajo. Esto se debe a que este algoritmo no necesita entrenamiento, sino que todo el trabajo lo hace al realizar una predicción.

De forma general, la manera en que este algoritmo funciona es utilizando el conjunto de datos de entrenamiento los cuales se usan como base para calcular la distancia entre un punto nuevo (predicción) y lo que se guardó. Pese a su buen rendimiento en tanto tiempo de predicción, entrenamiento y métricas, este modelo puede presentar problemas en cuanto a sobreajuste y carga computacional para cantidad de datos muy grandes.[48] Cabe recalcar que la razón por la cual tiene tiempos de predicciones tan rápidos es porque solo le fue pasado un dato, sin embargo, de ser pasados una cantidad más grande sus tiempos podrían volverse peores que los del modelo *ExtraTreesGini*. Además, a comparación del modelo *KneighborsUnif*, este no asigna una ponderación uniforme a todos los vecinos de un dato, sino que utiliza la distancia entre los puntos para darle un peso a estos. Es decir, que si alrededor di un punto para predecir tengo cinco otros puntos, 2 de ellos de una categoría y 3 de ellos de otra, el algoritmo realizaría su predicción en base de los pesos de cada uno de los vecinos.[49] Debido a esto, no se puede tomar el tiempo de predicción como un estimador real en caso se quieran predecir conjuntos más grandes al mismo tiempo. Asimismo, hay que tomar en cuenta que, pese a que sus métricas sean altas, es decir que logra clasificar ambas categorías correctamente, el hecho de utilizar el conjunto de datos de entrenamiento para realizar sus predicciones puede llevar a un sobreajuste del modelo.

Por otro lado, el *XGBoost*, presenta resultados de desempeño altos muy similares a los dos modelos de *Kneighbors*, sin embargo, su tiempo de entrenamiento es por mucho más alto que el de estos otros modelos. Esto sugiere que, aunque XGBoost tiene un excelente rendimiento en términos de *precision*, *recall* y *F1-Score*, la eficiencia computacional es un factor importante al seleccionar un modelo para aplicaciones prácticas. Sin embargo, XGBoost puede ser escalable para cantidades de datos grandes, lo cual determina que en un momento podría llegar a ser más rápido que *Kneighbors*, al menos para la predicción. Además, debido a sus parámetros de regularización y su arquitectura de *Decision Trees*, le permite generalizar más en sus predicciones y evitar el sobreajuste para el modelo de entrenamiento.[50] Esto es demostrado al observar la Figura 6, donde se puede ver que sus métricas son altas tanto para *recall* como para *F1-Score*, lo cual indica que logra clasificar correctamente ambas categorías.

En los resultados expresados en la Figura 9, se pueden observar la importancia de las características del conjunto de datos de prueba cuando los modelos generados por el

Autogluon realizan su predicción. En el gráfico se puede observar que las características que son más importante para la predicción son MFCC 2, 1, 10 y las menos importantes son las características relacionadas con el espectro, exceptuando *spectral bandwith*. Por otro lado, el hecho que todas las características MFCC tengan un puntaje alto indica que las características relacionadas con frecuencia del audio son importantes en la distinción de categorías. Incluso, dado que todas las características tienen un puntaje positivo, se puede determinar que todas aportan a la predicción del modelo y no son perjudiciales [51]. Sin embargo, se podrían eliminar aquellas características que tienen un aporte marginal como lo son las características relacionadas con el espectro de frecuencias, exceptuando *spectral bandwith.* Esto ayudaría a reducir la dimensionalidad del modelo, logrando que existan menos posibilidades de sobreajuste y tiempos más rápidos de entrenamiento y predicción al reducir la cantidad de datos que se tienen que procesar [52].

Por último, para revisar los resultados del método de filtros estadísticos, se compararon diferentes filtros como: filtro de media, filtro de mediana, filtro *low-pass*, filtro *high-pass,* filtro *band-pass* y filtro *notch*. Como se puede observar en la Figura 10, los filtros de media y mediana fueron los que mejor desempeño tuvieron en comparación a los resultados sin filtros. Esto se debe a que, al aplicar los filtros, las medias totales de los marcos de tiempo por cada característica de los quedaron casi iguales que sin filtros; por lo tanto, los modelos tuvieron un rendimiento similar. Por otro lado, el filtro *notch* tuvo el mejor rendimiento de los filtros restantes. Este filtro solo deja pasar datos, en los extremos, lo contrario a un filtro *band-pass*. Esto llega a indicar que los modelos utilizan los datos en los extremos de cada una de las características para poder diferenciar entre categorías y llevar a cabo su predicción. Esto es reforzado al observar el rendimiento de los modelos a utilizar el filtro *band-pass* el cual tuvo el peor rendimiento de todos los filtros. Cabe resaltar que estos filtros fueron aplicados una vez se tuvieron las características, sin embargo, también se pueden aplicar a señales crudas y se pueden obtener distintos resultados de ello. Aunque todos los modelos hayan tenido un peor desempeño con los datos con filtros, es importante señalar que estos modelos fueron entrenados en datos sin filtros. Pese a los resultados, se pudo utilizar esta información para entender cuáles son los datos en los que se fijan los modelos para realizar sus predicciones.

Como segundo método estudiado se realizó una comparación entre modelos *Long Short-Term Memory* con diferentes hiperparámetros. Como se puede observar en el Cuadro 7, se realizaron diferentes pruebas cambiando el hiperparámetro de paso de aprendizaje para el optimizador ADAM. Los resultados demostraron que el paso de aprendizaje de 0.0005 tuvo las mejores métricas, aunque no por mucho. Se debe tomar en cuenta que los tiempos obtenidos en los resultados fueron logrados utilizando Google Colab con acceso a una GPU, y por lo tanto sería más intensivo computacionalmente entrenar a esta red neuronal utilizando tan solo un CPU, en especial si se quieren comparar varios modelos. Comparando con modelos anteriores que lograron obtener un rendimiento parecido, pero con tiempos de entrenamiento más bajos, el modelo LSTM resulta muy excesivo en cuanto utilización de recursos.

Por otro lado, los pasos más pequeños demostraron tener las métricas más bajas también, además, su tiempo de entrenamiento fue más pequeño debido a que se detuvieron por *Early Stopping*. Esto demuestra que su bajo desempeño no se debe a falta de

entrenamiento, sino que simplemente que ya no habría mejora sin importar cuanto tiempo pasara. Dado que los resultados para los pasos de aprendizaje de 0.0001, 0.0005 y 0.001 son tan similares tanto para tiempo y métricas, tal como se muestra en la Figura 11, se debe obtener un balance entre el sobreajuste y las predicciones correctas. Utilizando las métricas de *recall* y *F1-Score*, se puede observar que el de 0.0005 tuvo un puntaje mayor que los otros dos, esto da a entender que si logra identificar correctamente entre *bonafide* y *deepfake*.

Como último método para comparar, se observó la habilidad de los humanos para reconocer entre voces reales y voces generadas por IA. Al observar Figura 12, los resultados muestran que en promedio las personas pueden acertar 5 de 10 preguntas correctamente, tal y como se demarca por la línea de media. Asimismo, todas aquellas preguntas que tienen un porcentaje de acierto arriba de la media pertenecen a la categoría *deepfake*. Esto demuestra que, en promedio, le es más fácil a las personas identificar voces generadas por IA. De tal manera, no se puede observar ninguna tendencia en el porcentaje de aciertos por cada pregunta. Por lo tanto, se puede decir que no hay suficiente información para determinar que los humanos mejoran su habilidad de discernimiento entre voces a medida que avanzaban en el examen.

Al realizar el examen se les pidió a los participantes que llenaran datos personales como edad, género y profesión. Como se puede observar en el Cuadro 8, el grupo de edades que mejor rendimiento tuvo en el examen fue el de 66 años o más. Sin embargo, al tomar en cuenta la distribución de edades que se puede ver en la Figura 13, ese grupo de edad tan solo conforma el 1% de las personas que hicieron el examen. Dado que aproximadamente 100 personas hicieron el examen, esto significa que el dato proviene tan solo de una persona. Por lo tanto, no es suficiente información como para determinar que ese grupo de personas tuvo el mayor porcentaje de aciertos. De forma contraria, el grupo con mayor porcentaje de participantes fue el más joven, de 18 a 25 años, además, también obtuvieron el puntaje más alto en las métricas. Dicho esto, se puede determinar que dentro de los grupos estudiados y considerando el porcentaje de personas en cada grupo, el grupo de 18 a 25 años tiene la mejor habilidad para distinguir entre voces generadas por IA y voces reales. Cabe resaltar que debido a que la mayor cantidad de personas eran de este grupo, los resultados obtenidos pueden ser más representativos que los de otros grupos.

En base a las respuestas obtenidas en la sección de profesión, se agruparon los resultados por cada grupo disponible. Como se puede observar en el Cuadro 9, la profesión que mejores métricas obtuvo fue la de Ingeniería/Tecnología con un puntaje de aproximadamente 0.59. Por otro lado, otras profesiones que mejores puntajes obtuvieron fueron los trabajos donde se relacionan directamente con otras personas, administración y negocios y el grupo de estudiantes. Además, como se puede ver en la Figura 14, estos cuatro grupos también conforman la mayoría de las respuestas en la pregunta de profesión. Esto sugiere que aquellas profesiones que tienen alguna interacción con personas o bien donde utilizan la tecnología tienden a poder distinguir mejor entre las dos categorías de voces. Además, tomando en cuenta que la mayoría de los que eligieron la opción de estudiantes también se encuentran dentro del grupo de más jóvenes, esto sugiere que este grupo esta familiarizado con la tecnología y por ello tienen buena capacidad para distinguir las voces. Dicho esto, cabe mencionar que los resultados obtenidos por los grupos son tan solo un

poco más altos que la media general. Por esto mismo, no se puede decir que ningún grupo en general, ya sea basado en profesión o edad, tenga una alta capacidad de poder distinguir entre una voz generada por IA y una real.

Por último, se compararon los resultados del examen de reconocimiento humano con el desempeño de los modelos de *machine learning* y redes neuronales utilizando el mismo conjunto de datos. Como se puede observar en la Figura 15, la mayoría de los modelos lograron obtener un puntaje perfecto en el conjunto de datos del examen, muy por arriba de las métricas obtenidas por los humanos. Sin embargo, el modelo LSTM, no logró obtener un puntaje perfecto pese a ser el modelo que tenía las mejores métricas en fases anteriores. Esto sugiere que el modelo sufrió de sobreajuste al conjunto de datos de entrenamiento o bien que simplemente no es tan bueno identificando estas voces. No obstante, algunos modelos con mejor rendimiento y más eficientes en cuanto a tiempo y utilización de recursos demostraron ser mejores en identificar las voces utilizadas para el examen. Esto denota que utilizar una red neuronal LSTM para este conjunto de datos podría ser más de lo necesario debido a que incurre en mayores gastos computacionales, de tiempo y de dinero al necesitar una GPU para entrenarse.

Se pudo determinar que modelos como el *KneighborsDist* serían los óptimos para este problema en particular al poder generalizar correctamente en cuanto a las dos categorías, incluso en audios que humanos no pueden reconocer tan bien. Además, cuentan con un tiempo de entrenamiento muy corto al no necesitar entrenarse como tal y pueden realizar la predicción de un dato de manera rápida. Aunque este modelo pertenecía al método de utilizar filtros estadísticos para mejorar las métricas obtenidas, este no fue el caso ya que el modelo obtuvo mejores resultados en el conjunto de datos sin ningún filtro aplicado. Además, el hecho de que no haya que optimizar hiperparámetros hace el proceso de creación del modelo sea aún más rápido. Cabe mencionar que modelos como LSTM y XGBoost podrían funcionar mejor en diferentes situaciones o incluso con un conjunto de datos distinto. Sin embargo, estos dos métodos consumen muchos más recursos computacionales que el modelo *KneighborsDist.*

En resumen, se pudo determinar que el método que mejor funciona para este conjunto de datos fueron los algoritmos de *machine learning* sin ningún tipo de filtro aplicado. Dentro de los algoritmos estudiados, el *KneighborsDist* de la librería de Autogluon en Python fue el que mejor se adaptó a los requerimientos de uso de recursos, métricas y tiempo de entrenamiento. Por otro lado, el método que peores resultados obtuvo fue el de reconocimiento humano, lo cual indica que en general los humanos no son tan buenos distinguiendo entre voces generadas por IA y voces reales. Esto sugiere que la utilización de alguna herramienta que utilice un modelo de los anteriormente planteados podría ser útil en escenarios donde se necesite una detección rápida y eficiente de voces generadas por inteligencia artificial.

Es importante mencionar que las características extraídas para este conjunto de datos pueden ser reducidas al ver su importancia en la predicción y de esta manera mejorar el tiempo de entrenamiento y de predicción de modelos más complejos. Incluso, de esta manera se podría eliminar cierta parte del sobreajuste que presento el modelo LSTM y llegaría a volverse el mejor modelo para diferenciar entre las dos categorías de voces.

VIII. Conclusiones

El objetivo de este estudio fue identificar el método más preciso para distinguir entre voces reales y generadas por IA, evaluando modelos de *machine learning* con filtros estadísticos aplicados, redes neuronales y la identificación humana. Para ello, se analizó la eficiencia de cada método en recursos y precisión, el potencial de mejora en la identificación humana, y los parámetros óptimos para redes neuronales y modelos de *machine learning*

1. Con base a los resultados obtenidos y a la comparación de cada método, se pudo determinar que el método más preciso y el que tuvo mejor puntaje en las métricas de *recall, precision, accuracy* y *F1-score* para lograr clasificar una voz verdadera y una voz generada por inteligencia artificial es fue la red neuronal que utiliza una arquitectura LSTM.
2. En cuanto a recursos utilizados, en el área de entrenamiento el más económico sería el método que utiliza Filtros estadísticos, en específico el modelo *KneighborsDist* proporcionado por la librería Autogluon. Al no necesitar un entrenamiento como tal, se obtuvieron el tiempo más bajo en ese aspecto.
3. Al observar los resultados obtenidos del método de reconocimiento humano se pudo determinar que en general, las respuestas obtenidas no presentan una mejora a lo largo del examen, sino que son preguntas en específico las que son contestadas con mayor precisión.
4. Al realizar el entrenamiento de la red neuronal, el parámetro que más afecto fue el *learning rate* o paso de aprendizaje. Se pudo determinar que un paso de aprendizaje de 0.0005 con el optimizador ADAM obtuvo los mejores puntajes en todas las métricas de predicción.
5. Utilizando la librería de Autogluon se pudo determinar que las características extraídas que más afectan a la predicción de los modelos que utilizan filtros estadísticos son MFCC 2, MFCC1 y MFCC 10.
6. Para el método de *machine learning* que utilizaba filtros estadísticos, se pudo determinar que el modelo de *KneighborsDist* del Autogluon presentaba los mejores resultados al tener parámetro que asignaba un peso a la distancia entre puntos de datos.
7. Para el modelo de machine learning SVM del método que utilizaba filtros estadísticos, los mejore parámetros fueron utilizar un kernel radial con un coeficiente de penalización C igual a 1000.

8. Para el modelo de *machine learning* que utilizaba filtros estadísticos, se pudo determinar que no aplicar ningún tipo de filtro al conjunto de datos presentaba los mejore resultados.
9. El grupo de personas que mejor rendimiento tuvo en el examen de reconocimiento humano con respecto a su edad fue el de 18 a 25 años.
10. El grupo de personas que mejor rendimiento tuvo en el examen de reconocimiento humano con respecto a su profesión fue el de ingeniería/tecnología.
11. Se pudo encontrar que, aunque la arquitectura LSTM presentaba las mejores métricas en cuanto a predicciones, al utilizar el conjunto de datos del examen de reconocimiento humano no tuvo tan buen rendimiento como otros modelos, lo cual indica que existe algún tipo de sobreajuste.

IX. Bibliografía

[1] A. Bajaj, “Voice Cloning Using Artificial Intelligence Algorithm — RNN,” Medium. Accessed: May 23, 2024. [Online]. Available: https://aryanbajaj13.medium.com/voice-cloning-using-artificial-intelligence-algorithm-rnn-3ad56c39e7dc

[2] Microsoft Prensa, “El aumento de la adopción de la tecnología de Inteligencia Artificial (IA) genera expectación y pone de relieve la importancia de las conversaciones familiares sobre la seguridad online, según un nuevo estudio de Microsoft,” Microsoft. Accessed: May 23, 2024. [Online]. Available: https://news.microsoft.com/es-es/2024/02/06/el-aumento-de-la-adopcion-de-la-tecnologia-de-inteligencia-artificial-ia-genera-expectacion-y-pone-de-relieve-la-importancia-de-las-conversaciones-familiares-sobre-la-seguridad-online-segun-un-nuev/

[3] Silverio Mario, “ChatGPT: número de usuarios y estadísticas,” PrimeWeb. Accessed: May 23, 2024. [Online]. Available: https://www.primeweb.com.mx/chatgpt-usuarios-estadisticas#:~:text=y%20mucho%20m%C3%A1s.-,Estad%C3%ADsticas%20clave%20de%20ChatGPT,de%20visitantes%20durante%20febrero%202024.

[4] Martina, “IA vs. industria musical: El auge del clonado de voz mediante IA,” iMusician. Accessed: May 23, 2024. [Online]. Available: https://imusician.pro/es/recursos-practicos/blog/el-auge-del-clonado-de-voz-mediante-ia

[5] K. Waserman, “Keeping up with scammers: Deepfake voice fraud,” TheStatement. Accessed: May 23, 2024. [Online]. Available: https://thestatement.bokf.com/articles/2023/09/keeping-up-with-scammers

[6] M. Casado, “La cara auditiva: El reconocimiento de las personas a través de la voz,” CienciaCognitiva. Accessed: May 23, 2024. [Online]. Available: https://www.cienciacognitiva.org/?p=854

[7] A. Cuartero, “Bud Bunny estalla contra la IA que ha suplantado su voz en una canción viral: ‘Ustedes no merecen ser mis amigos,’” LaVanguardia. Accessed: May 23, 2024. [Online]. Available:

https://www.lavanguardia.com/gente/20231107/9358860/bud-bunny-estalla-ia-suplantado-voz-cancion-viral-merecen-mis-amigos.html

[8] A. Bunn, "Artificial Imposters—Cybercriminals Turn to AI Voice Cloning for a New Breed of Scam," McAfee. Accessed: May 23, 2024. [Online]. Available: https://www.mcafee.com/blogs/privacy-identity-protection/artificial-imposters-cybercriminals-turn-to-ai-voice-cloning-for-a-new-breed-of-scam/

[9] S. Ionescu, "El mejor software de clonación de voz por IA en 2024," Geekflare. Accessed: Jun. 26, 2024. [Online]. Available: https://geekflare.com/es/best-ai-voice-cloning-software/

[10] A. Andreu, "Microsoft crea una herramienta para clonar cualquier voz: con solo escuchar un audio de 3 segundos, VALL-E es capaz de hablar como tú," BusinessInsider. Accessed: May 23, 2024. [Online]. Available: https://www.businessinsider.es/vall-ia-microsoft-clonar-cualquier-voz-1182236

[11] S. Russel and P. Norvig, *Artificial Intelligence A Modern Approach Fourth Edition*, Cuarta edición. Hoboken: Pearson, 2021.

[12] A. Kaplan and M. Haenlein, "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence," *Bus Horiz*, vol. 62, no. 1, pp. 15–25, Jan. 2019, doi: 10.1016/J.BUSHOR.2018.08.004.

[13] A. Esteva *et al.*, "A guide to deep learning in healthcare," Jan. 01, 2019, *Nature Publishing Group*. doi: 10.1038/s41591-018-0316-z.

[14] "The future of work after COVID-19," 2021. Accessed: Jun. 03, 2024. [Online]. Available: https://www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-after-covid-19

[15] M. Chui, R. Roberts, and L. Yee, "Generative AI is here: How tools like ChatGPT could change your business," Dec. 2022.

[16] B. G. Acosta-Enriquez, M. A. Arbulú Ballesteros, O. Huamaní Jordan, C. López Roca, and K. Saavedra Tirado, "Analysis of college students' attitudes toward the use of ChatGPT in their academic activities: effect of intent to use, verification of information and responsible use," *BMC Psychol*, vol. 12, no. 1, Dec. 2024, doi: 10.1186/s40359-024-01764-z.

[17] I. S. Gabashvili, "Systematic Review The impact and applications of ChatGPT: a Systematic Review of Literature Reviews", doi: 10.17605/OSF.IO/87U6Q.

[18] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," Jul. 01, 2021, *Association for Computing Machinery*. doi: 10.1145/3457607.

[19] M. Westerlund, "The Emergence of Deepfake Technology: A Review," *Technology Innovation Management Review*, vol. 9, no. 11, pp. 39–52, Nov. 2019.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” [Online]. Available: http://code.google.com/p/cuda-convnet/

[21] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661

[22] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” May 27, 2015, *Nature Publishing Group*. doi: 10.1038/nature14539.

[23] M. Nayak, “Demystifying the Architecture of Long Short Term Memory (LSTM) Networks,” Towards Ai. Accessed: Aug. 19, 2024. [Online]. Available: https://towardsai.net/p/l/demystifying-the-architecture-of-long-short-term-memory-lstm-networks

[24] S. Hochreiter and J. ¨ Urgen Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[25] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” *IEEE Trans Neural Netw Learn Syst*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[26] K. Barua, A. Rahim, S. Parizat, A. Noor, and M. Jannah, “Voice Impersonation Detection using LSTM based RNN and Explainable AI,” Brac University, Bangladesh, 2021.

[27] D. Efanov, P. Aleksandrov, and N. Karapetyants, “The BiLSTM-based synthesized speech recognition,” in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 415–421. doi: 10.1016/j.procs.2022.11.086.

[28] S. Y. Lim, D. K. Chae, and S. C. Lee, “Detecting Deepfake Voice Using Explainable Deep Learning Techniques,” *Applied Sciences (Switzerland)*, vol. 12, no. 8, Apr. 2022, doi: 10.3390/app12083926.

[29] L. R. . Rabiner and R. W. . Schafer, *Theory and applications of digital speech processing*. Pearson, 2011.

[30] G. F. . Harris and P. A. . Smith, *Human motion analysis : current applications and future directions*. Institute of Electrical and Electronics Engineers, 1996.

[31] S. W. Smith, *Digital Signal Processing Second Edition*, Second edition. San Diego: California Technical Publishing, 1999. [Online]. Available: www.DSPguide.com

[32] Bernard. Gold, Nelson. Morgan, and D. P. W. . Ellis, *Speech and audio signal processing : processing and perception of speech and music*. Wiley-Blackwell, 2011.

[33] S. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Trans Acoust*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[34] T. Ganchev, N. Fakotakis, and G. Kokkinakis, “Comparative evaluation of various MFCC implementations on the speaker verification task,” 2005. [Online]. Available: https://www.researchgate.net/publication/228756314

[35] M. Todisco, H. Delgado, and N. Evans, “Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification,” *Comput Speech Lang*, vol. 45, pp. 516–535, Sep. 2017, doi: 10.1016/J.CSL.2017.01.001.

[36] Pierre. Duhamel and Michel. Kieffer, *Joint source-channel decoding : a cross-layer perspective with applications in video broadcasting over mobile and wireless networks*. Academic Press/Elsevier, 2010.

[37] K. Stevens, *Acoustic Phonetics*, vol. 30. Massachusetts: The MIT press, 2000.

[38] J. G. . Proakis and D. G. . Manolakis, *Digital signal processing*. Pearson, 2014.

[39] C. N. Babu and B. E. Reddy, “A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data,” *Appl Soft Comput*, vol. 23, pp. 27–38, Oct. 2014, doi: 10.1016/j.asoc.2014.05.028.

[40] E. Ataman, V. K. Aatre, and Wong K. M., “Some Statistical Properties of Median Filters,” *IEEE Trans Acoust*, vol. ASSP-29, no. 5, pp. 1073–1075, Oct. 1981.

[41] W. Yost, *FUNDAMENTALS OF HEARING WILLI*, Fourth Edition. Chicago: Academic Press, 2000.

[42] B. Goldstein, *Sensation and Perception*, Ninth Edition. Wadsworth: Cengage Learning, 2013. [Online]. Available: www.cengagebrain.com

[43] B. C. J. Moore, *An Introduction to the Psychology of Hearing Sixth Edition*, Sixth edition. Cambridge: Brill, 2013.

[44] J. Pickles, *An Introduction to the Physiology of Hearing Fourth Edition*, Fourth edition. Emerald Group Publishing LImited, 2012.

[45] H. Fastl and E. Zwicker, *Springer Series in Information Sciences*, Third Edition. Berlin: Springer, 2007.

[46] J. Katz, M. Chasin, K. English, L. Hood, and K. Tillery, Eds., *HANDBOOK OF CLINICAL AUDIOLOGY*, Seventh Edition. Philadelphia: Wolters Kluwer, 2015.

[47] H. Delgado, “ASVspoof 2021 Challenge - Speech Deepfake Database,” ASV, May 2021. doi: 10.5281/zenodo.4835108.

[48] A. Tharwat, “Parameter investigation of support vector machine classifier with kernel functions,” *Knowl Inf Syst*, vol. 61, no. 3, pp. 1269–1302, Dec. 2019, doi: 10.1007/s10115-019-01335-4.

[49] scikit-learn developers, “Nearest Neighbors,” Scikit-Learn. Accessed: Oct. 03, 2024. [Online]. Available: https://scikit-learn.org/1.5/modules/neighbors.html#id5

[50] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[51] N. Erickson *et al.*, “AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data,” Mar. 2020.

[52] M. A. Salam, A. T. Azar, M. S. Elgendy, and K. M. Fouad, “The Effect of Different Dimensionality Reduction Techniques on Machine Learning Overfitting Problem,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 641–655, 2021, doi: 10.14569/IJACSA.2021.0120480.