ENFOQUE DE APRENDIZAJE DE DISTANCIAS PARA LA DETECCIÓN DE CAMBIOS DE MICRÓFONO EN ARCHIVOS DE AUDIO

ELKIN FABIAN CALDERÓN ARDILA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2023

ENFOQUE DE APRENDIZAJE DE DISTANCIAS PARA LA DETECCIÓN DE CAMBIOS DE MICRÓFONO EN ARCHIVOS DE AUDIO

ELKIN FABIAN CALDERÓN ARDILA

Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director
Franklin Alexander Sepúlveda Sepúlveda
PhD en Ingeniería Automática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
2023

DEDICATORIA

A mi madre María Inés, mi hermano Rolando y mi hermana Diana, quienes siempre me han brindado su apoyo incondicional en todo.

A la memoria de mi padre, quien siempre fue mi mayor ejemplo de determinación y perseverancia.

AGRADECIMIENTOS

A todos los docentes que me acompañaron y guiaron durante mi proceso de formación, especialmente a mi director de trabajo de grado, Dr. Franklin Alexander Sepúlveda.

A mis amigos y compañeros de estudio, con quienes tuve la dicha de vivir este proceso y cada una de sus vicisitudes.

A la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E3T) y a la Universidad Industrial de Santander (UIS).

CONTENIDO

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

RESUMEN

TÍTULO: ENFOQUE DE APRENDIZAJE DE DISTANCIAS PARA LA DETECCIÓN DE CAMBIOS DE MICRÓFONO EN ARCHIVOS DE AUDIO [*]

AUTOR: ELKIN FABIAN CALDERÓN ARDILA [**]

PALABRAS CLAVE: ALTERACIÓN DE AUDIO, APRENDIZAJE DE MÉTRICAS DE DISTANCIA, DISTANCIA DE MAHALANOBIS, LMNN, NCA, ITML, DIVERGENCIA DE JENSEN-SHANNON, COEFICIENTES CEPSTRALES EN LAS FRECUENCIAS DE MEL.

DESCRIPCIÓN:

En el presente trabajo se lleva a cabo la implementación y evaluación de dos algoritmos para la detección de cambios o inconsistencias de micrófono en grabaciones de audio para aplicaciones forenses mediante un enfoque basado en el aprendizaje de distancias.

El primer algoritmo consiste en el cálculo de la Distancia de Mahalanobis mediante los métodos LMNN (Large Margin Nearest Neighbor), NCA (Neighbourhood Components Analysis) e ITML (Information Theoretic Metric Learning) a partir de vectores de información espectral extraídos de ventanas de audio. Estos vectores se obtienen del cálculo de los MFCC (Coeficientes Cepstrales en las Frecuencias de Mel), omitiendo el último paso que consiste en la aplicación de la DCT (Transformada de Coseno Discreta) y, utilizando un banco de 128 filtros en la escala mel con límites entre 0 y 8000 Hz. Por otro lado, el segundo algoritmo consiste en hallar la Divergencia de Jensen-Shannon mayor entre los conjuntos de filtros número 1 a 6 y, 99 a 128 del banco de filtros en escala mel mencionado. Para la evaluación de estos algoritmos se utiliza la base de datos AVSpoof, creada por Ergünay et al. El análisis de los resultados tanto individuales como en conjunto obtenidos tras la implementación de estos dos algoritmos se realiza mediante tres diferentes tipos de clasificadores: SVM (máquinas de vectores de soporte), ANN (redes neuronales artificiales) y k-NN (k vecinos más cercanos).

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Físicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Franklin Alexander Sepúlveda Sepúlveda, PhD. en Ingeniería Automática.

ABSTRACT

TITLE: DISTANCE LEARNING APPROACH FOR MICROPHONE SPLICING DETECTION IN AUDIO FILES [*]

AUTHOR: ELKIN FABIAN CALDERÓN ARDILA [**]

KEYWORDS: AUDIO SPLICING, DISTANCE METRIC LEARNING, MAHALANOBIS DISTANCE, LMNN, NCA, ITML, JENSEN-SHANNON DIVERGENCE, MEL FREQUENCY CEPSTRAL COEFFICIENTS.

DESCRIPTION:

In this project, a metric learning-based approach is used to implement and evaluate two algorithms for detecting inconsistencies regarding the microphone in audio recordings, which is intended for forensic applications.

The first algorithm consists of calculating the Mahalanobis Distance using three methods: LMNN (Large Margin Nearest Neighbor), NCA (Neighbourhood Components Analysis), and ITML (Information Theoretic Metric Learning). This calculation is carried out from spectral information vectors extracted from audio windows; these vectors are obtained by computing the MFCC (Mel Frequency Cepstral Coefficients), excluding the final step involving the implementation of the DCT (Discrete Cosine Transform) and applying a 128 filters bank on the mel scale. The frequency limits of this filters bank are 0 and 8000 Hz. On the other hand, the second algorithm consists of finding the greatest Jensen-Shannon Divergence between the sets of filters number 1 to 6 and 99 to 128 of the mel scale filters bank previously mentioned.

The evaluation of these algorithms is performed using the database AVSpoof, which was created by Ergünay et al. The analysis of results, both individual and overall, obtained after the algorithms implementation is done using three different classifiers: SVM (Support Vector Machines), ANN (Artificial neural network) and k-NN (k-Nearest Neighbors).

---

[*] Bachelor Thesis

[**] Facultad de Ingenierías Físicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Franklin Alexander Sepúlveda Sepúlveda, PhD. en Ingeniería Automática.

INTRODUCCIÓN

En Colombia y otros países la identificación de la voz como prueba pericial dentro del sistema penal judicial es de gran importancia en las investigaciones de crímenes tales como secuestro, extorsión, narcotráfico y terrorismo. Ello se debe a que el desarrollo de los mismos implica el uso de canales de comunicación como celulares y teléfonos, cuyas grabaciones de audio son comúnmente utilizadas como material probatorio en procesos judiciales por este tipo de crímenes [1].

Los audios a utilizar en procedimientos que implican tareas de verificación del hablante (cotejo de voces en el contexto colombiano) poseen propiedades que podrían degradar la efectividad de los sistemas y procedimientos de verificación; por consiguiente, antes de realizar cualquier tipo de procedimiento de cotejo de voces típicamente se solicita garantizar que el audio correspondiente cumpla ciertas características que lo cataloguen como apto para ser utilizado como evidencia en un proceso judicial [2]. Una de las características que se deben probar es que el audio no haya sido adulterado, por ejemplo, mediante la inserción de segmentos de audio.

Dentro de los procedimientos de engaño que implican la manipulación de segmentos de audio se menciona: a) Inserción, que implica insertar segmentos cortos de audio en un audio de mayor tamaño, que puede provenir del mismo audio o de otro audio, pero del mismo hablante. b) Sustitución, que implica reemplazar un segmento corto de audio por

---

[1] Fiscalia General de la Nación. República de Colombia. *Manual Único de Criminalística*.

[2] L. Romito y V. Galatà. “Towards a protocol in speaker recognition analysis”. En: *Forensic Science International* 146, Supplement (2004). Mediterranean Academy of Forensic Sciences 1st Workshop, S107 -S111. DOI: http://dx.doi.org/10.1016/j.forsciint.2004.09.033.

otro. c) Eliminación, que corresponde a borrar segmentos de audio.

Aunque existen herramientas que permiten detectar cortes e inserciones en registros de audio tales como *EdiTracker*, este está basado en detección de cortes de fase y cambios abruptos de frecuencias en componentes armónicos previamente seleccionados por el experto a cargo del cotejo de voces [3]. De manera algo similar, se reporta el uso de métodos basados en la detección de discontinuidades en la fase de la señal inducida por la red eléctrica sobre dispositivos electrónicos; sin embargo, el uso de estos está limitado por el acceso a esta señal de referencia a ser proveída por las empresas de generación y distribución de la energía eléctrica [4]. Adicionalmente, en experimentos previos realizados por el autor del presente trabajo se encontró que la señal de $60\ Hz$ de la red eléctrica no se refleja en todos los registros de audio, especialmente si provienen de conversaciones de celular. Esta misma observación también se reporta en algunos trabajos del estado del arte [5].

Otros métodos se enfocan en detectar inconsistencias y cambios en los sonidos y ruido de fondo [6]; o, en la detección de inconsistencias en la reverberación [7]. En el presente

---

[3] Speech Technology Center, EdiTracker, Manual de usuario, En: `https://speechpro.com/files/product/ikarlab2/docs/editracker_ug_eng.pdf`, 2012.

[4] P. A. A. Esquef, J. A. Apolinário y L. W. P. Biscainho. "Edit Detection in Speech Recordings via Instantaneous Electric Network Frequency Variations". En: *IEEE Transactions on Information Forensics and Security* 9.12 (dic. de 2014), págs. 2314-2326.

[5] S. Gupta, S. Cho y C. C. J. Kuo. "Current Developments and Future Trends in Audio Authentication". En: *IEEE MultiMedia* 19.1 (ene. de 2012), págs. 50-59. DOI: 10.1109/MMUL.2011.74.

[6] Hong Zhao et al. "Audio splicing detection and localization using environmental signature". En: *Multimedia Tools and Applications* 76 (2017), 13897–13927. DOI: 10.1007/s11042-016-3758-7.

[7] Davide Capoferri et al. "Speech Audio Splicing Detection and Localization Exploiting Reverberation Cues". En: *2020 IEEE International Workshop on Information Forensics and Security (WIFS)* (2020), págs. 1-6.

trabajo se plantea utilizar el enfoque de detección de inconsistencias en el micrófono con el fin de detectar inserción de segmentos de audio que provienen de otro registro de audio pero que corresponden al mismo hablante y que se desarrollan en el mismo ambiente. En particular, se implementa y se pone a prueba un enfoque basado en el aprendizaje de distancias [8] [9] del tipo Mahalanobis y del tipo probabilístico. Una vez se tiene un conjunto de distancias estas ingresan a modo de entradas a un clasificador a fin de detectar inconsistencias o cambios de micrófono en grabaciones de audio.

Distancias. En el presente trabajo se asume que durante el proceso de recortado e inserción se podrían generar inconsistencias en la huella del dispositivo con el que se realizaron los audios, las cuales se reflejen en distancias espectrales de mayor tamaño entre tramos consecutivos de audio. Sin embargo, para ello se requiere tener algoritmos adecuados de medición de distancias.

Se define la distancia $d(x, y)$ entre dos representaciones espectrales x y y correspondientes a dos segmentos de voz localizados en tiempos t_+ y t_-, respectivamente; donde t es el tiempo en cuya vecindad se busca analizar la presencia de inconsistencias de micrófono. A esta distancia se le denomina métrica de distancia si cumple con las siguientes propiedades[8] [9]:

- *Simetría*. La distancia de x a y es igual a la distancia de y a x. El requisito de simetría asegura que una medida de distancia entre dos tramos de sonido no distinga entre

---

[8] Michel Marie Deza y Elena Deza. *Encyclopedia of Distances*. 2nd ed. Springer, 2013. DOI: https://doi.org/10.1007/978-3-642-30958-8.

[9] Juan Luis Suárez, Salvador García y Francisco Herrera. "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges". En: *Neurocomputing* 425 (2021), págs. 300-322. DOI: https://doi.org/10.1016/j.neucom.2020.08.017.

cuál es una referencia y cuál es un sonido de prueba.

- *No negatividad*. La distancia entre x y y es positiva, excepto en el caso en que $x = y$. En este caso la distancia es cero.
- *Desigualdad triangular.* $d(x, z) \leq d(x, y) + d(y, z)$, para representaciones espectrales de segmentos de voz localizados en tiempos t_x, t_y y t_z.

1. OBJETIVOS

Objetivo general

- Evaluar dos algoritmos para la detección de cambios de micrófono en grabaciones de audio mediante un enfoque basado en el aprendizaje de métricas de distancia.

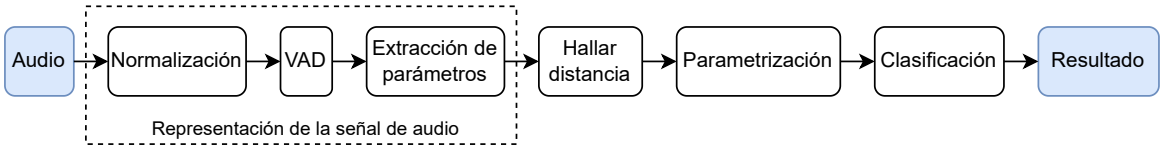
Objetivos específicos

- Implementar un algoritmo que utiliza ejemplos de puntos similares o disimilares en $\mathbf{R}^n$ para aprender una métrica de distancia Euclidiana que respeta estas relaciones y que permita la detección de cambios de micrófono en grabaciones de audio.
- Implementar un algoritmo basado en la estimación de distancias probabilistas entre dos ventanas consecutivas de archivos de audio.
- Evaluar tanto individual como conjuntamente los algoritmos implementados, mediante un enfoque basado en el aprendizaje de métricas de distancia y la implementación de un clasificador.

2. MÉTODO

En la figura 1 se muestran las etapas de funcionamiento del sistema una vez se tienen tanto las distancias como el clasificador entrenados.

Figura 1. Diagrama general del método propuesto



Una vez entrenado el sistema, este tiene como entrada la señal de audio, luego, en cada etapa la información acústica es procesada.

2.1. REPRESENTACIÓN DE LA SEÑAL DE AUDIO

2.1.1. Pre-procesamiento Antes de proceder a obtener un conjunto de mediciones que representan, en este caso, el espectro del segmento de voz, se procede a realizar un procedimiento de normalización y de detección de voz. El primero es un procedimiento estándar que busca eliminar la influencia de la diferencia en el volumen en los diferentes registros de audio. Luego, la etapa de detección de voz busca determinar aquellas porciones de la señal en la cual, en efecto, sí hay voz al tiempo que descarta silencios y porciones donde no hay voz pero sí hay señal acústica.

Normalización. Tras realizar la lectura de los archivos de audio a usar en la fase de entrenamiento, se procede a realizar la normalización de estos, esto se realiza restando su media y dividiendo en su desviación estándar. De esta forma se obtienen registros de

audio con media cero y desviación estándar uno y, se evita que características tales como una diferente amplitud de la señal para diferentes dispositivos de grabación intervengan más adelante en la correcta clasificación llevada a cabo por los algoritmos.

Detección de actividad de voz. Se evidenció que las características extraídas a partir de tramos de silencio no eran significativas para la identificación de los diferentes micrófonos, por el contrario, hacía este trabajo más difícil y menos preciso. Por este motivo, se procedió a realizar la detección de actividad de voz (VAD, por sus siglas en inglés) en todos los registros de audio a analizar para posteriormente realizar la remoción de silencios de estos. Esto se hizo mediante la función *detectSpeech* de MATLAB.

Durante el desarrollo del presente trabajo se observó que la variabilidad de la longitud de los tramos de silencios influía directamente en la extracción de los parámetros, y los datos en solo silencios y con ventanas de análisis pequeñas se veían afectados por el porcentaje de audio incorrectamente catalogado como silencio.

2.1.2. Representación espectral en la escala mel La mayoría de los algoritmos dependen de que se les dé unos buenos parámetros en sus entradas para realizar una correcta clasificación o para evitar que el usuario deba sintonizarlos manualmente. Por tal motivo, una de las primeras decisiones en cualquier sistema de clasificación es escoger qué características usar para representar la señal que se busca clasificar, para de esta forma facilitar el trabajo de los algoritmos de aprendizaje automático.

Teniendo en cuenta lo anterior, el primer paso para abordar el problema de detección de inconsistencias de micrófono en archivos de audio es determinar una representación adecuada de la señal acústica. Al investigar esto en el estado del arte, se evidenció que en el campo del procesamiento de voz y audio, uno de los parámetros que se han venido utilizando en mayor medida, y que han mostrado ampliamente su eficiencia son los Co-

eficientes Cepstrales en la escala Mel de Frecuencias (MFCC, Mel-Frequency Cepstral Coeeficients). Los MFCC han sido utilizados ampliamente en aplicaciones de voz, pero más recientemente también en el campo de la identificación de fuentes de grabación de audio [10].

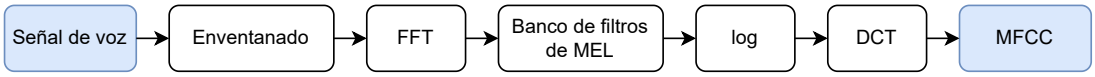
Para hallar estos coeficientes se toma la señal de voz y se somete a un banco de filtros triangulares pasa banda traslapados, los cuales poseen un mismo ancho de banda, pero en la escala mel, obteniéndose de esta forma valores de energía asociados a cada rango de las frecuencias de estos filtros.

La escala de mel refleja de una mejor manera la forma en la que los seres humanos perciben el sonido, es decir, diferenciando mejor las frecuencias bajas. Esta escala es aproximadamente lineal por debajo de 1 kHz y logarítmica sobre este valor. El cálculo de la escala de mel a partir de la frecuencia *f* en Hz se realiza según la ecuación (1) .

$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right) \tag{1}$$

Los pasos para la extracción de los MFCC se muestran a continuación en la figura 2.

Figura 2. Pasos para extraer los MFCC



El primer paso es separar la señal en pequeños tramos con un tamaño y valor de paso determinado. Posteriormente se realiza el envenenado de la señal, generalmente usando una ventana del tipo Hamming; el propósito de esto es evitar las discontinuidades al

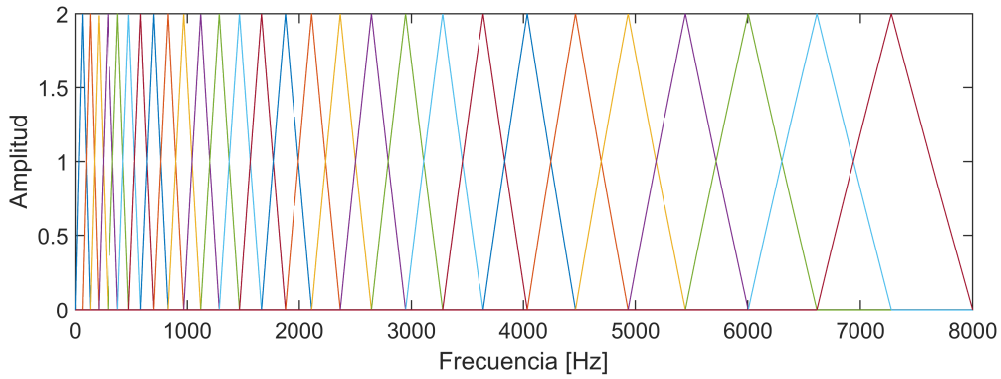
[10] Daniel Garcia-Romero y Carol Y. Espy-Wilson. "Automatic acquisition device identification from speech recordings". En: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, págs. 1806-1809. DOI: 10.1109/ICASSP.2010.5495407.

principio y al final de cada uno de los tramos a analizar. A continuación se presenta la ecuación de la ventana tipo Hamming.

$$v\left(n\right)=0.54-0.46\cos\left(\frac{2\pi n}{N}\right),0\leq n\leq N \tag{2}$$

El siguiente paso es obtener la representación espectral de la señal usando la FFT de la señal enventanada. Luego, se aplica el banco de filtros triangulares en la escala mel al espectro obtenido en el paso anterior y se hallan las energías en cada filtro. En la figura 3 a continuación se muestra como ejemplo un banco de 28 filtros, en esta se evidencia el comportamiento logarítmico de la escala mel.

Figura 3. Banco de filtros en escala mel



Finalmente, a fin de obtener los MFCCs se halla el logaritmo natural de las energías de cada filtro en la escala de frecuencias de mel y se aplica la Transformada de Coseno Discreta a estos valores logarítmicos. El cálculo de los MFCC se realizó mediante la función melcepts de la *Toolbox VOICEBOX* de MATLAB [11], sin embargo, en el presente trabajo no se tuvo en cuenta la última etapa de aplicar la DCT, para lo cual fue necesario modificar

[11] Mike Brookes. *VOICEBOX: Speech Processing Toolbox for MATLAB*. `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`.

la mencionada función melcepts.

En el presente trabajo se decidió utilizar un banco con 128 filtros. Como Anexo A se encuentra la tabla 8 con las frecuencias en Hz del banco de filtros en escala mel utilizado.

2.2. DISTANCIAS PROBABILÍSTICAS

Una forma de detectar inconsistencias en el micrófono dentro de registros de audio es mediante el uso de distancias de tipo probabilístico, entre las cuales se pueden mencionar la Divergencia de Kullback-Liebler y la Divergencia de Jensen–Shannon.

2.2.1. Divergencia de Kullback-Liebler Sea X un conjunto de observaciones en un segmento de audio A localizado en $t_- < t$ y Y un conjunto de observaciones del segmento de audio B localizado en $t_+ > t$. La divergencia estadística de *Kullback-Liebler* entre los segmentos A y B se obtiene al comparar sus correspondientes funciones de probabilidad $f_X(z)$ y $f_Y(z)$ de la forma,

$$d_{KL}(X \parallel Y) = \int_z f_X(z) \log \frac{f_X(z)}{f_Y(z)} \tag{3}$$

La unidad de medida es nats si se utiliza el logaritmo natural y bits si se usa el logaritmo en base dos. La divergencia de Kullback-Liebler es una métrica estadística que mide como una función de probabilidad $f_Y(z)$ difiere de una función de probabilidad de referencia $f_X(z)$. El cálculo con funciones de probabilidad continua es más complejo que para el caso discreto, por lo cual se plantea el uso de versiones discretas de las funciones de probabilidad: $p_X(z)$ y $p_Y(z)$.

$$d_{KL}(X \parallel Y) = \sum_{z_i} p_X(z_i) \log \frac{p_X(z_i)}{p_Y(z_i)} \tag{4}$$

La divergencia de Kullback-Leibler no es considerada una distancia porque no cumple con el requisito de simetría[8], ya que

$$d_{KL}(X \parallel Y) \neq d_{KL}(Y \parallel X) \tag{5}$$

2.2.2. Divergencia de Jensen-Shannon Con base en la divergencia de Kullback-Liebler se puede definir la divergencia de Jensen-Shannon, la cual a diferencia de la primera, sí es simétrica y por lo tanto se considera como una verdadera distancia métrica. La divergencia de Jensen-Shannon está dada por la ecuación 6,

$$d_{JS}(p_X \parallel p_Y) = \frac{1}{2} d_{KL}\Big(p_X(z) \parallel q(z)\Big) + \frac{1}{2} d_{KL}\Big(p_Y(z) \parallel q(z)\Big) \tag{6}$$

donde q es la distribución de probabilidad promedio de las distribuciones p_X y p_Y dada por:

$$q = \frac{1}{2}\left(p_X + p_Y\right) \tag{7}$$

Sin embargo, al tratar de estimar $p_X(x)$ y $p_Y(y)$ aparece el problema de la maldición de la dimensionalidad debido a que X y Y son vectores espectrales de dimensión p y la cantidad de observaciones disponibles está dada por la cantidad de ventanas de análisis dentro de los segmentos a compararse. Por lo cual, en su lugar se plantea tomar $d_{KL}(X, Y)$ como el máximo de entre las posibles dimensiones de la forma,

$$d^*_{KL}(X, Y) = \max_i \left\{ d_{KL}(x_i, y_i) \right\} \tag{8}$$

donde $x = [x_1\, x_2\, \cdots\, x_p]$ y $y = [y_1\, y_2\, \cdots\, y_p]$; y, p denota el tamaño del vector espectral para cada ventana de análisis.

2.3. DISTANCIAS DE MAHALANOBIS

Existen varias formas de definir métricas de distancia. Una de ellas se origina al considerar una transformación lineal de la representación espectral x de la forma $A \cdot x$, con lo cual resulta una distancia de la forma,

$$\begin{aligned} d^2(x, y) &= (x - y)^\intercal M(x - y) \\ &= (Ax - Ay)^\intercal (Ax - Ay) \end{aligned} \tag{9}$$

donde $M = A^\intercal \cdot A$, en cuyo caso M es una matriz definida positiva. Esta distancia se llama distancia de Mahalanobis, la cual es utilizada para determinar el grado de similitud entre dos variables aleatorias multidimensionales y tiene en cuenta la correlación entre estas variables.

El área de aprendizaje de distancias se centra en determinar aquellos valores de la matriz M que resulten adecuados para medir cuantitativamente la similitud o diferencia entre un conjunto de datos dados. En el caso de este proyecto, el enfoque de aprendizaje de distancias se utiliza para detectar inconsistencias de micrófono dentro de registros de audio.

Dentro de los algoritmos que permiten estimar la matriz M para la distancia definida en (9) se encuentran: LMNN (Large Margin Nearest Neighbor Classification), NCA (Neighborhood Component Analysis) e ITML (Information Theoretic Metric Learning), propuestos por por Kilian Q. Weinberger et al. [12], Jacob Goldberger et al. [13] y Jason V. Davis et

---

[12] Kilian Q Weinberger, John Blitzer y Lawrence K. Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". En: *Advances in Neural Information Processing Systems 18*. Ed. por Y. Weiss, B. Schölkopf y J. C. Platt. MIT Press, 2006, págs. 1473-1480.

[13] Jacob Goldberger et al. "Neighbourhood Components Analysis". En: *Advances in Neural Information Processing Systems.* Ed. por L. Saul, Y. Weiss y L. Bottou. Vol. 17. MIT Press, 2004.

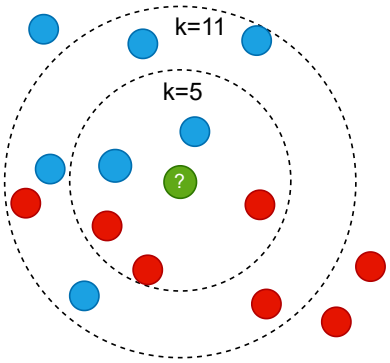
al. [14], respectivamente.

2.3.1. Distancia Euclideana Para el caso en el que A corresponde a la matriz identidad, la distancia d de la ecuación (9) se reduce a la distancia Euclidiana d_E a continuación, en donde n es el número de dimensiones de los vectores x y y.

$$d_E(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{10}$$

La distancia Euclidiana se utiliza en el algoritmo de los k vecinos más cercanos, el cual es un algoritmo de clasificación supervisada cuyo propósito es clasificar datos con clase desconocida utilizando un conjunto de ejemplos con clases ya conocidas. Para hacer esto, el algoritmo calcula la distancia entre el elemento a clasificar y los ejemplos del conjunto, selecciona los k elementos con menor distancia y clasifica el dato dado en la clase que más se repite entre esos k elementos.

Figura 4. Representación del algoritmo de k vecinos más cercanos



En la figura 4 se encuentra una representación de este algoritmo. Se tiene que para $k = 5$

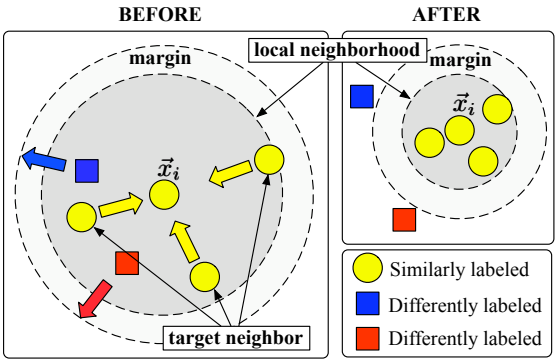
[14] Jason V. Davis et al. "Information-Theoretic Metric Learning". En: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, 209–216. DOI: 10.1145/1273496.1273523.

el círculo verde sería clasificado como rojo, ya que hay 3 rojos y 2 azules en el círculo interno, mientras que para $k = 11$, este sería clasificado como azul. ya que hay 6 azules y 5 rojos en el círculo externo.

2.3.2. Large Margin Nearest Neighbor Classification El algoritmo k-NN descrito previamente puede ser significativamente mejorado tras aprender una métrica de distancia de ejemplos con etiquetas. Con el propósito de lograr esto, el algoritmo LMNN aprende una métrica de distancia Mahalanobis descrita en la ecuación (9) con el objetivo de que los k vecinos más cercanos pertenezcan a la misma clase mientras que los ejemplos de clases diferentes se separen por una gran margen [12].

En la figura 5 (tomada de [12]) a continuación se encuentra la representación del algoritmo.

Figura 5. Representación del algoritmo LMNN



Fuente: 12.

El algoritmo LMNN define el problema como una instancia de programación semidefinida en la que se penalizan las distancias grandes entre cada entrada y sus vecinos que comparten la misma clase y, también se penalizan las distancias cortas entre cada entrada y sus vecinos de diferente clase. El problema se resuelve al minimizar la ecuación:

$$s(k) = \sum_{ij} \eta_{ij} (x_i - x_j)^T M (x_i - x_j) + c \sum_{ij} \eta_{ij} (1 - y_{il}) \xi_{ijl} \tag{11}$$

sujeta a:

$$\begin{aligned} (x_i - x_l)^T M (x_i - x_l) - (x_i - x_j)^T M (x_i - x_j) &\geq 1 - \xi_{ijl} \\ \xi_{ijl} &\geq 0 \\ M &\succeq 0 \end{aligned} \tag{12}$$

donde c es una constante positiva, ξ_{ij} son variables de holgura y η_{ij} tiene valor 0 o 1 e indica si una entrada x_j es un vecino de la misma clase que x_i cuya distancia se desea minimizar.

2.3.3. Neighborhood Component Analysis Suponiendo un punto x_i, y dada la ecuación (9), se tiene la distancia $d_A(x_i, x_j) = \|Ax_i - Ax_j\|^2$; pero, $d_A(x_i, x_j)$ es muy susceptible a cambios de A y pequeños cambios de A produce grandes cambios en $d_A(x_i, x_j)$. En su lugar se utiliza una función de la forma,

$$p_{ij} = \frac{e^{\|Ax_i - Ax_j\|}}{\sum_{k \neq i} e^{\|Ax_i - Ax_k\|}} \tag{13}$$

la cual produce valores de 0 a 1; y por tanto, p_{ij} puede ser usado para determinar la probabilidad de que i sea clasificado correctamente. Al realizar la sumatoria de la expresión p_{ij} sobre aquellos puntos j pertenecientes a la clase q ($j \in C_q$), si este i está muy alejado de los puntos $j \in C_q$, el valor total de $\sum_{j \in C_q} = p_{ij}$ será un valor muy pequeño, denotando una baja probabilidad de pertenencia a esta clase.

En tal sentido, la función objetivo a formularse sería maximizar la cantidad esperada de

puntos clasificados correctamente, es decir,

$$\max_A g(A) = \max_A \sum_i p_i = \max_A \sum_i \sum_{j \in C_q} p_{ij} \tag{14}$$

Para resolver este problema se utilizan algoritmos iterativos, e.g. algoritmos de primera derivada, para lo cual se requiere el gradiente dado por

$$\frac{\partial}{\partial A} g = 2A \sum_i \left(p_i \sum_k p_{ik} x_{ik} x_{ik}^\intercal - \sum_{j \in C_q} p_{ij} x_{ij} x_{ij}^\intercal \right) \tag{15}$$

2.3.4. Information Theoretic Metric Learning Dos puntos (x_i, x_j) son similares si la distancia entre ellos es más pequeña que un umbral superior dado, es decir, si se cumple que $d_A(x_i, x_j) \leq u$ para un valor u suficientemente pequeño. De manera similar, dos puntos son disímiles si $d_A(x_i, x_j) \geq l$ para un valor l suficientemente grande. Para casos de clasificación en los que se conocen las etiquetas de clase para cada ejemplo, los puntos que pertenecen a una misma clase se consideran similares y, puntos de clases diferentes son considerados disímiles. Las inecuaciones recién descritas sirven de restricciones al problema de optimización que permite la inferencia de la matriz definida positiva M que describe la distancia de Mahalanobis en la ecuación (9).

La cantidad de posibilidades para la matriz M es infinita; pero, se aplica regularización tratando de llevar M (que genera $d_M()$) hacia una forma más simple M_0. Esta matriz M_0 podría ser la matriz identidad I (al reemplazar I en (9) se llega a la distancia Euclidiana). Se utiliza el criterio de información de KL descrito en la sección 2.2.1 para medir la distancia entre las funciones de probabilidad de estas dos distancias de la forma [14],

$$KL(f_X(x; M_0) \mid\mid f_X(x; M)) = \int f_X(x; M_0) \log \left(\frac{f_X(x; M_0)}{f_X(x; M)} \right) dx \tag{16}$$

Al establecer los dos conjuntos de índices de datos S (similares) y D (índices de datos disímiles), el problema de optimización que permite inferir M es,

$$\min_{M} \quad KL\Big(f_X(x; M_0) \mid\mid f_X(x; M)\Big) \tag{17}$$

$$\begin{aligned} \text{sujeto a} \quad & d_M(x_i, x_j) \leq u \quad (i,j) \in \mathrm{S} \\ & d_M(x_i, x_j) \geq l \quad (i,j) \in \mathrm{D} \\ & M \succeq 0 \end{aligned} \tag{18}$$

2.4. CLASIFICACIÓN

Para llevar a cabo la clasificación entre audios con inconsistencias en el micrófono y audios sin estas, se debe en primera medida realizar la extracción de características a cada unión de archivos de audio y, posteriormente, hacer uso de un clasificador.

Parametrización. Al aplicar las distancias sobre ventanas consecutivas de análisis que se desplazan en pasos de $32\ ms$ se obtienen varios vectores de distancias (un vector por cada distancia). La longitud del vector de distancia hallado en ambos algoritmos propuestos es directamente proporcional a la longitud de cada unión de archivos de audio, por consiguiente, varía de tamaño y hace que los vectores de distancias no sean directamente comparables entre sí. Por tal motivo, los parámetros a usar en el clasificador no son directamente los vectores de distancias obtenidos, sino otros parámetros finales que se obtienen a partir de estos tal como se describe a continuación:

Dif(d) **:** El primer parámetro es la diferencia entre el valor máximo de distancia y el valor mínimo de esta. La unión de dos archivos de audio de diferente micrófono ocasionan un pico en la distancia y por consiguiente un valor mayor de esta diferencia.

$$\text{Dif}(d) = d_{\text{max}} - d_{\text{min}} \tag{19}$$

$\varepsilon(d)$**:** La entropía es un concepto que proviene de la teoría de la información. En particular, la entropía de *Shannon* es un concepto que aplica a variables aleatorias, especialmente a discretas.

$$H(x) = \sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{20}$$

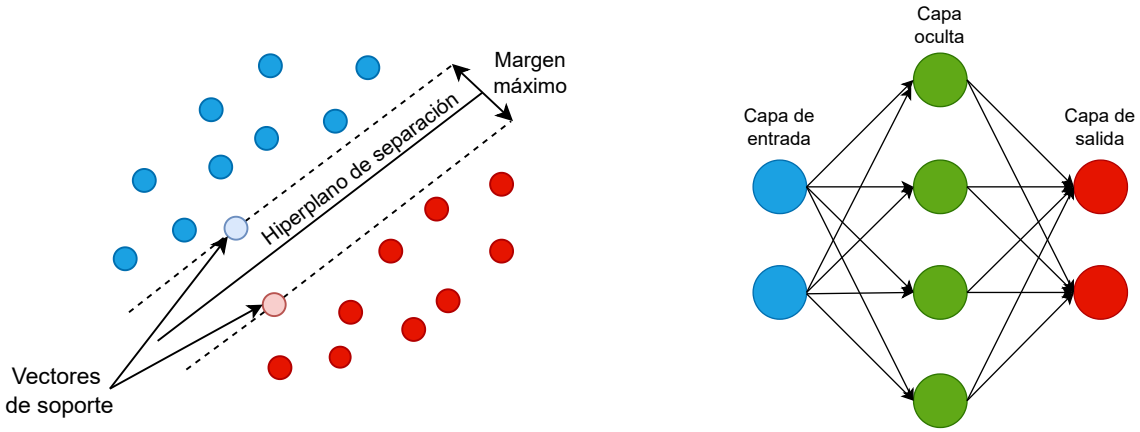
Este concepto se utiliza en la teoría de ondículas (*wavelets*) y, en el presente trabajo se observó mejoría en los resultados al utilizar la función de cálculo de entropía de la *Toolbox* de ondículas de MATLAB de nombre *wentropy*. En particular, se observó que cuando no hay inconsistencias en el micrófono el valor de entropía tiende a ser mayor; en contraste, cuando sí hay inconsistencias, el valor de entropía tiende a ser menor.

Clasificadores. En el presente trabajo se utilizaron los clasificadores máquinas de vectores de soporte, k vecinos más cercanos y redes neuronales artificiales para evaluar los algoritmos implementados. La explicación del funcionamiento del algoritmo de clasificación de los k vecinos más cercanos se encuentra en la sección 2.3.1. A continuación se realiza una breve explicación de los otros dos tipos de clasificadores utilizados.

SVM: Las máquinas de vectores de soporte son un algoritmo de aprendizaje supervisado ampliamente utilizado para clasificación y regresión. Este algoritmo toma un conjunto de datos etiquetados con clases, los ubica en un espacio multidimensional y busca separar las clases lo más ampliamente posible a través de un hiperplano de separación que maximiza la distancia entre los datos más cercanos de diferentes clases, recibiendo estos puntos el nombre de vectores de soporte. Tras hacer esto, se pueden clasificar nuevos datos según la ubicación en la que se encuentren dentro del espacio multidimensional respecto del hiperplano. En la figura 6 se encuentra una representación de este clasificador.

ANN: Las redes neuronales artificiales, ANN por sus siglas en inglés, son un modelo de aprendizaje automático inspirado en el funcionamiento del cerebro humano. Están compuestas por un conjunto interconectado de nodos o neuronas, las cuales reciben una o varias entradas, realizan un cálculo a partir de estas y generan una o varias salidas, las cuales pueden estar conectadas a otras neuronas formando una estructura de capas y conexiones en la red. Estas conexiones tienen unos pesos asociados, los cuales se van ajustando mientras la red neuronal se entrena intentando minimizar una función de pérdida. Una vez entrenada, la red neuronal puede utilizarse para clasificar nuevos datos.

Figura 6. Representaciones de un clasificador tipo SVM y de una red neuronal artificial: (a) Clasificador tipo SVM; (b) Red neuronal artificial



2.5. EVALUACIÓN DEL SISTEMA

La evaluación de los algoritmos implementados se realiza en base a varias métricas que son calculadas a partir de la matriz de confusión, la cual se presenta a continuación en la figura 7. En esta matriz de confusión se encuentran los dos tipos de errores que tiene un sistema de detección de cambios: el error tipo I, el cual ocurre si un cambio verdadero no

es hallado, y el error tipo II, que se produce cuando un cambio detectado no corresponde a un cambio real (falsa alarma).

Figura 7. Matriz de confusión

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativos	Falsos positivos (FP)	Verdaderos negativos (VN)

Las siguientes relaciones son las métricas que se obtienen a partir de la matriz de confusión:

Precisión: Es la proporción de verdaderos positivos entre el número total de resultados positivos.

$$\text{Precisión} = \frac{\text{VP}}{\text{VP + FP}} \tag{21}$$

Recall: También llamada sensibilidad, es la proporción de verdaderos positivos en comparación con el total de positivos presentes en los datos.

$$\text{Recall} = \frac{\text{VP}}{\text{VP + FN}} \tag{22}$$

Valor F: Es una métrica de evaluación que combina la precisión y el recall en una sola puntuación para medir el rendimiento de un clasificador. Tiene en cuenta tanto la capacidad del clasificador para evitar clasificar incorrectamente ejemplos negativos como positivos, como su capacidad para clasificar correctamente los ejemplos positivos. El Valor F varía de 0 a 1, siendo un valor más próximo a 1 una indicación de

un mejor desempeño del algoritmo.

$$\text{Valor F} = \frac{2 \cdot \text{Precisión} \cdot \text{Sensibilidad}}{\text{Precisión + Sensibilidad}} \tag{23}$$

Exactitud: Es la proporción de resultados verdaderos entre el número total de casos examinados.

$$\text{Exactitud} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{FN} + \text{VN}} \tag{24}$$

Para llevar a cabo el análisis de los resultados de los algoritmos implementados se hará uso de la matriz de confusión y sus métricas relacionadas: precisión, recall, valor F y exactitud.

La forma en la que se obtienen los cambios a detectar es mediante la unión de dos archivos de audio del mismo hablante y sección grabados con diferente micrófono pertenecientes a la base de datos AVSpoof [15].

[15] S. K. Ergünay et al. "On the vulnerability of speaker verification to realistic voice spoofing". En: *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. Sep. de 2015, págs. 1-6. DOI: 10.1109/BTAS.2015.7358783.

3. RESULTADOS

3.1. CONJUNTO DE DATOS

Para evaluar el desempeño tanto de las distancias como del conjunto de los dos algoritmos implementados se utilizó la base de datos de nombre *AVSpoof* creada por S. K. Ergünay et al.[15], la cual consta de archivos de audio grabados con tres dispositivos diferentes (Iphone 3GS, Samsumg Galaxy S4 y micrófono de buena calidad AT2020USB+) y con pronunciaciones tanto dependientes como independientes del texto. Esta base de datos fue creada con el propósito de permitir a investigadores evaluar algoritmos realizados para detectar distintos tipos de modificaciones realizadas en archivos de audio, tales como re-play, síntesis y conversión de voz. En el presente trabajo se escogió utilizar esta base de datos porque supera a otras en aspectos clave para el caso analizado en este trabajo, tales como el número de sesiones de grabación y el número de dispositivos de grabación [16].

La base de datos se compone de registros de audio pronunciados por 44 hablantes, 31 hombres y 13 mujeres, los cuales fueron recopilados en 4 diferentes sesiones llevadas a cabo en días diferentes y en donde se efectuaron variaciones en características tales como el ruido de fondo y la reverberación. Además, con el objetivo de estandarizar la configuración de las grabaciones, se fijó la posición de los dispositivos de grabación para cada sesión y hablante. La frecuencia de muestreo de los archivos en esta base de datos es de $16\ kHz$.

---

[16] Zhizheng Wu et al. “SAS: A speaker verification spoofing database containing diverse attacks”. En: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, págs. 4440-4444. DOI: 10.1109/ICASSP.2015.7178810.

En cada sesión los participantes se sometieron a tres diferentes protocolos de adquisición, tal como se enumeran a continuación:

- *Read*: El participante lee entre 10 y 40 oraciones predefinidas.
- *Pass*: El participante lee 5 sentencias cortas.
- *Free*: El participante habla con libertad sobre cualquier tema.

Durante el entrenamiento y la evaluación se usaron archivos tipo *read*. Se decidió no utilizar archivos de audio del tipo *pass* ya que estos eran muy cortos, en la mayoría de los casos tenían una longitud inferior a $500\ ms$, valor menor a los $1024\ ms$ de tamaño de las ventanas de análisis usadas.

Los archivos de audio de cada hablante se utilizaron en solo una de las dos fases del proceso, entrenamiento y evaluación. En la tabla 1 a continuación se relaciona en qué fase del proceso se utilizaron los archivos de audio de cada hablante:

Tabla 1. Hablantes de la base de datos

	Hablantes
Entrenamiento	m001, m002, m004, m005, m006, m009, m010, m013, m014, m015, m016, m017, m021, m022, m026, m044, f003, f007, f008, f018, f019, f020, f023
Evaluación	m011, m012, m027, m029, m030, m033, m034, m035, m036, m037, m038, m039, m040, m041, m042, f024, f025, f028, f031, f032, f043.

3.2. DISTANCIA DE MAHALANOBIS

Para comenzar con el entrenamiento de esta distancia se parte de la base de datos descrita en la sección 3.1. Se toman 5 registros audio de cada uno de los 23 hablantes de la fase de entrenamiento (ver tabla 1), para cada una de las 4 sesiones de grabación y, por cada uno de los tres micrófonos, los cuales en adelante se denominarán *phone1*, *phone2* y *laptop*, tal como los denominaron los autores de la base de datos[15]. De esta forma de obtienen un total de 1380 registros de audio a ser utilizados en la fase de entrenamiento

de la distancia.

Se toma cada audio normalizado y sin silencios, estos se subdividen en ventanas de 1024 ms con pasos de 256 ms y se halla un vector de representación espectral de tamaño 128 para cada ventana, el cual consiste en el cálculo de una versión modificada de los MFCC documentados en la sección 2.1, en la que no se realiza el ultimo paso de hallar la transformada de coseno discreta (DCT por sus siglas en inglés). Tras esto se obtiene para cada registro de audio un conjunto de n vectores de tamaño 128 (correspondiente al número de filtros en escala mel utilizados), donde n depende del tamaño de cada registro de audio. Todo lo anterior se realiza con la función melcepts de la Toolbox Voicebox de MATLAB [11], en su versión modificada.

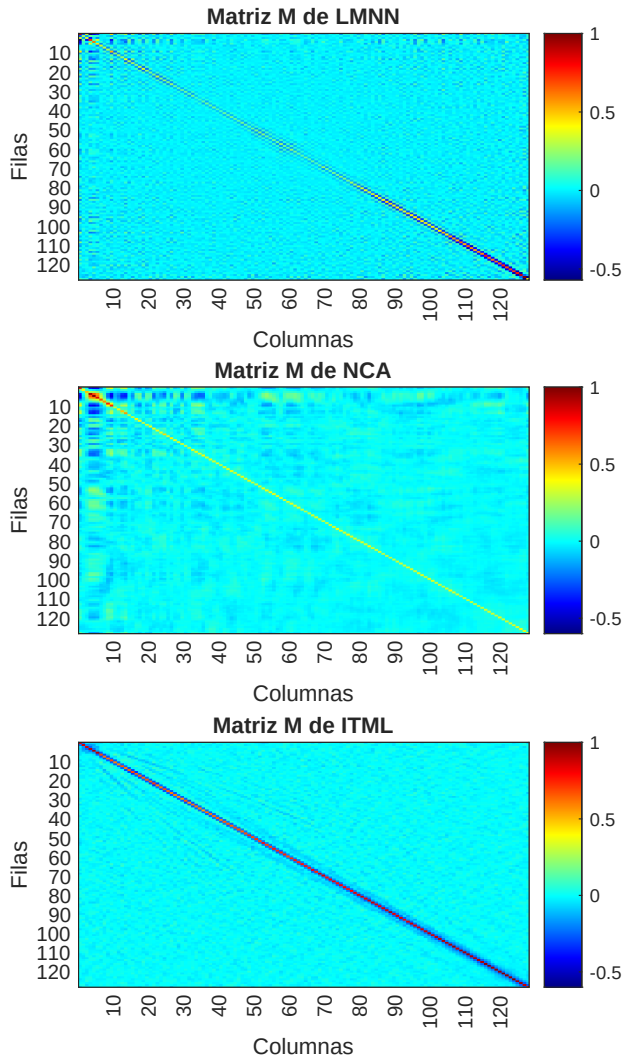
Luego, se toman aleatoriamente 3 de estos n vectores de características por cada archivo de audio de la fase de entrenamiento, obteniéndose de esta forma un total de 4140 vectores de representación espectral, los cuales posteriormente se utilizan para hallar las matrices M de la métrica de distancia de Mahalanobis descrita en la ecuación 9.

Las matrices M correspondientes a las métricas de distancia de Mahalanobis descritas en las secciones 2.3.2, 2.3.3 Y 2.3.4, llamadas LMNN, NCA e ITML se hallan utilizando el paquete *metric-learn* de *Python* [17]. Como modo de resultado se obtienen las matrices M para cada método. En la figura 8 se muestra el mapa de calor de cada una de las matrices de Mahalanobis M para cada uno de los tres métodos. Para la realización de estos mapas de calor se procedió a normalizar cada matriz M respecto a su entrada de máximo valor, esto se hizo para lograr que las gráficas fueran directamente comparables y facilitar su análisis.

---

[17] William de Vazelhes et al. “metric-learn: Metric Learning Algorithms in Python”. En: *Journal of Machine Learning Research* 21.138 (2020), págs. 1-6.

Figura 8. Mapa de calor de las matrices de Mahalanobis M para cada uno de los métodos LMNN, NCA e ITML.



En la figura 8 se puede observar que la mayor parte de los elementos de las matrices M correspondientes a la distancia de Mahalanobis para los tres métodos LMNN, NCA e ITML, tienen valores muy cercanos a cero y, por esta razón, su aporte a la distancia de la ecuación (9) es mucho menor respecto al aporte de los elementos de las diagonales. Con el propósito de ver más de cerca el comportamiento de estas diagonales, se presentan a continuación en la figura 9.

Figura 9. Diagonales de las matrices M normalizadas del algoritmo con distancias de Mahalanobis

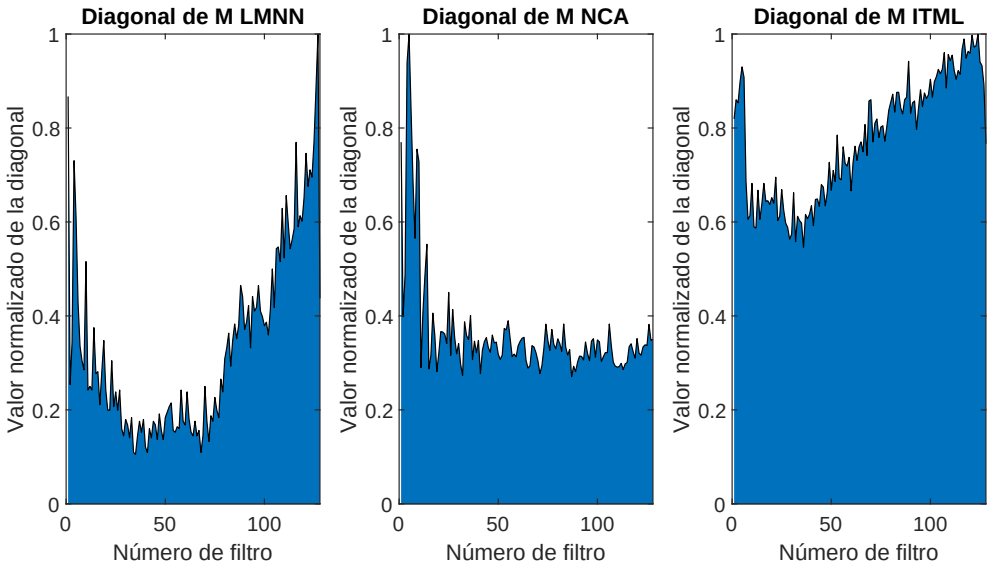
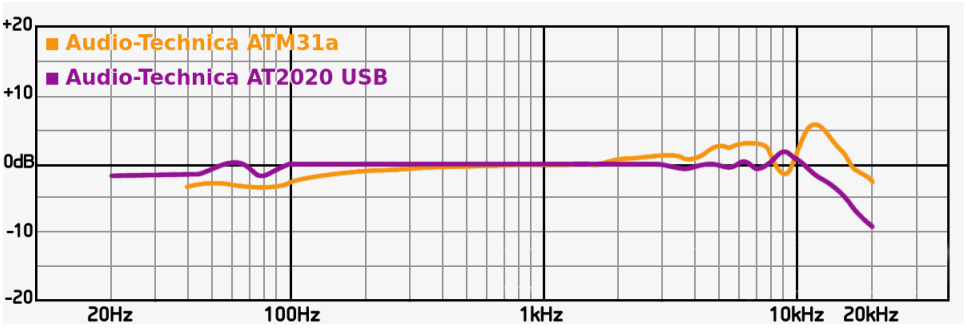


Figura 10. Respuesta en frecuencia de un par de micrófonos de referencias *AudioTechnica ATM31a* y *AudioTechnica ATM2020 USB*



En la figura 10 (adaptada de [18]) se encuentra la respuesta en frecuencia del micrófono *AT2020USB*, la cual es una referencia muy cercana a la del tercer micrófono con el que se grabaron los registros de audio de la base de datos utilizada en este trabajo (*AT2020USB+*), junto con la de otro micrófono de la misma marca y con una referencia

[18] Matt Mcglynn. *RecordingHacks, a microphone database and search engine*. `http://recordinghacks.com/`.

cercana. En esta gráfica se evidencia que la respuesta en frecuencia es plana en el rango de frecuencias de mayor importancia para el reconocimiento de voz, es decir, aproximadamente entre $100\ Hz$ y $3\ kHz$. Este es el rango en el cual los fabricantes de micrófonos se enfocan principalmente, descuidando los componentes de frecuencias mayores y menores a este.

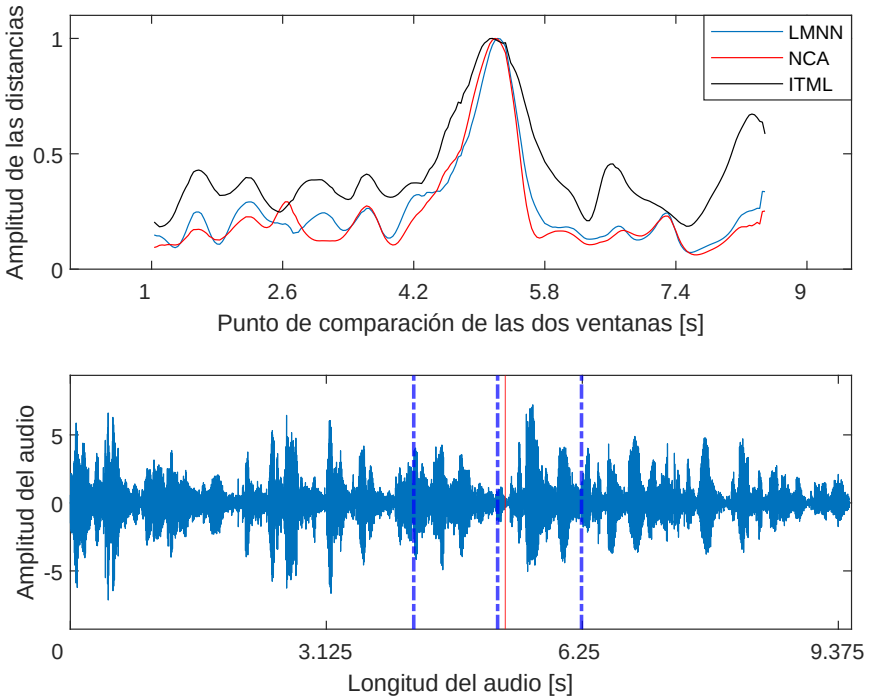
Al parecer, los micrófonos se diferencian entre sí en mayor medida en rangos de frecuencia menores a $100\ Hz$ y mayores a $3\ kHz$. Debido a esto, la contribución que realizan los componentes en frecuencias diferentes al rango habitual de la voz humana para efectuar la diferenciación entre un determinado micrófono y otro con una referencia distinta, es mayor a la que realizan los componentes en frecuencias de la voz.

Lo explicado anteriormente se evidencia de manera clara en la figura 9, donde se puede observar que el valor de los elementos de las diagonales de las matrices M correspondientes a rangos de frecuencia de la voz humana es menor respecto a aquellos elementos que corresponden a rangos de frecuencias más altas o bajas.

Para continuar con la fase de evaluación, en primera medida se seleccionan los archivos de audio que se utilizarán para esta fase. Se toman 10 registros de audio por cada uno de los 21 hablantes de la fase de evaluación (ver tabla 1) y para cada una de las 4 sesiones de grabación. Cada uno de estos archivos de audio se une con otro del mismo hablante y sesión, pero con micrófono igual o diferente, según los siguientes pares: $phone1 - phone1$, $phone2 - phone2$, $laptop - laptop$, $phone1 - laptop$, $phone2 - phone1$, $phone2 - laptop$. Cabe aclarar que de los hablantes $m011$ y $m012$ no se utilizaron archivos de audio de la sesión 1, debido a la falta de estos en la base de datos. Teniendo en cuenta lo anterior, finalmente se tiene un total de 4920 uniones de archivos de audio, en donde se sabe que la mitad de estas presentan inconsistencias de micrófono y la otra mitad no.

Ahora se procede a hallar la distancia de Mahalanobis correspondiente a cada una de las tres matrices M para cada par de registros de audio de la fase de evaluación. El primer paso para hallar los vectores de distancia es aplicar los pasos de pre-procesamiento ya descritos con anterioridad: normalización por media y desviación estándar y, VAD para eliminación de silencios. Luego, se divide la señal de audio en ventanas Z de tamaño $2048\ ms$ y esta a su vez en las ventanas X y Y de tamaño $1024\ ms$ (secciones izquierda y derecha de Z, respectivamente). Se procede a hallar los vectores de características espectrales de X y Y y, a aplicar la fórmula 9 para hallar el valor de distancia correspondiente a este par de ventanas. Tras esto se traslada la ventana Z un total de $32\ ms$ y se repite el proceso hasta llegar al final de la unión de los dos registros de audio en cuestión.

Figura 11. Vectores de distancia para el caso de inconsistencias en el micrófono, mediante distancia de Mahalanobis



En la figura 11 se presentan los vectores de distancia para cada uno de los tres métodos LMNN, NCA e ITML; estos vectores se obtienen tras analizar la unión de los archivos de audio correspondientes a la oración *01* de *phone1* y la oración *02* de *laptop* de la sesión *1* del hablante *m033*. Para realizar esta gráfica cada vector de distancia se normalizó respecto a su valor máximo. Esto se hizo para que todos tuvieran valor máximo 1 y se pudieran interpretar en conjunto con mayor facilidad. La línea roja horizontal en la gráfica inferior de la figura se ubica en el punto de unión de los dos archivos de audio, mientras que las líneas punteadas azules representan las ventanas de comparación en las cuales se presenta el pico de distancia mediante el método LMNN.

En la figura 11 se puede evidenciar que el valor máximo de distancia se da en un punto cercano a aquel en el cual las dos ventanas de comparación contienen mayormente información de uno de los dos micrófonos y, va disminuyendo lentamente a medida que las ventanas de comparación comienzan a contener información mezclada de ambos micrófonos.

Luego de hallar estas distancias se obtienen a partir de cada una de ellas los dos parámetros de clasificación explicados en la sección 2.4: Dif(d) , diferencia entre el valor máximo y mínimo de distancia, y $\varepsilon(d)$, la entropía de Shannon de la distancia utilizando la función *wentropy* de MATLAB. A partir de estos parámetros y utilizando tres tipos diferentes de clasificadores, en la sección 3.4 que se encuentra más adelante se muestran los resultados de la evaluación del algoritmo de la distancia de Mahalanobis.

3.3. DIVERGENCIA DE JENSEN-SHANNON

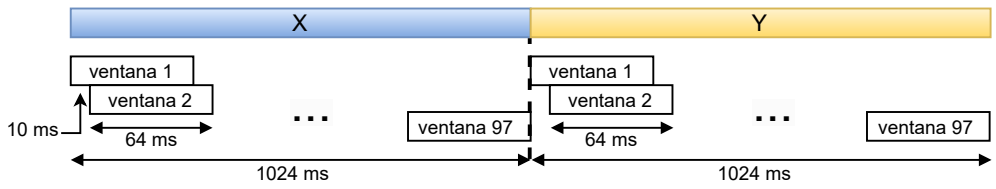
Para evaluar esta distancia, la cual se describe en la sección 2.2.2, se siguen los mismos pasos previos que para la de Mahalanobis. En primer lugar se realiza la lectura de

los archivos de audio, que son aquellos de los hablantes relacionados en la tabla 1 como usados en la fase de evaluación de la distancia de Mahalanobis. Luego, se realiza la normalización tal como en esa distancia, restando la media y dividiendo en la desviación estándar. Posteriormente, se aplica VAD mediante la función *detecSpeech* de MATLAB para eliminar silencios y, finalmente, se concatenan los archivos de audio.

Luego, el procedimiento es el siguiente:

- El archivo de audio se subdivide en ventanas grandes Z de $2048\ ms$ de tamaño, las cuales están compuestas por las ventanas adyacentes X y Y, cada una de $1024\ ms$. Luego, cada ventana X y Y se subdividen a su vez en ventanas pequeñas de $64\ ms$ con paso de $10\ ms$, tal como se observa en la figura 12.
- Se halla la función de probabilidad p_X a partir del conjunto de 97 ventanas en los que se subdivide X, y la función de probabilidad p_Y del conjunto de ventanas de Y. A partir de estas y haciendo uso de las fórmulas 6 y 7 se calcula el valor de distancia de Jensen Shannon d_{JS} que corresponde a la ventana Z. Esto se realiza por cada uno de los 128 elementos de los vectores de información espectral hallados.
- Finalmente, tras obtener los valores de d_{JS} correspondientes a la primera ventana Z, a esta se le realiza un desplazamiento de $32\ ms$, y se repite el proceso de manera sucesiva hasta el final del archivo.

Figura 12. Subdivisión de ventanas del algoritmo con distancia probabilística



Tras terminar los pasos anteriores, se tiene un total de 128 valores de d_{JS} por cada ventana Z, uno por cada filtro del banco de filtros utilizado. A fin de evitar la maldición de

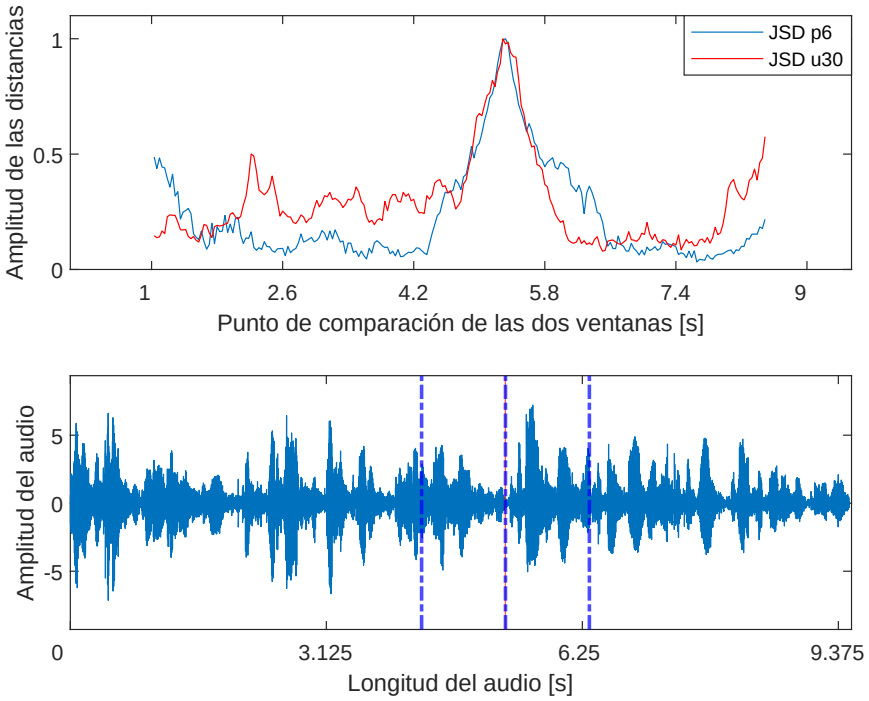
la dimensionalidad, como se expresa en 2.2.2, se procede a tomar el valor mayor de d_{JS} de entre las dimensiones de interés. Este procedimiento evita tener que estimar una función de probabilidad de dimensión p. Las dimensiones de interés corresponden a: los primeros 6 bancos de filtros y, los últimos 30 bancos de filtros, los cuales corresponden a las frecuencias menores que $103\ Hz$ y a las frecuencias entre $4048\ Hz$ y $8000\ Hz$, respectivamente (ver tabla 8). Estos rangos se escogieron en base al análisis efectuado anteriormente acerca de las diagonales de las matrices M de las métricas de distancias de Mahalanobis halladas a través de los métodos LMNN, NCA e ITML.

La notación $6p$ y $30u$ se utilizará desde este punto para indicar si un resultado o gráfica se refiere a la d_{JS} hallada a partir de los primeros 6 filtros del banco de filtros, o a partir de los últimos 30, respectivamente.

En la figura 13 a continuación se presentan los vectores de distancia correspondientes a los métodos $6p$ y $30u$ de la divergencia de Jensen-Shannon. Estos vectores se obtuvieron al analizar la misma unión de archivos utilizada para la figura (11). Se evidencia un comportamiento similar al observado en el caso de la distancia de Mahalanobis: el valor máximo de distancia se da en un punto cercano a aquel en el cual las dos ventanas de comparación contienen mayormente información de uno de los dos micrófonos y, va disminuyendo lentamente a medida que las estas ventanas comienzan a contener información mezclada de ambos micrófonos.

Luego de hallar estas distancias para cada par de archivos de la fase de evaluación, al igual que el caso de la distancia de Mahalanobis, se hallan los dos parámetros de clasificación. En la sección a continuación se muestran los resultados de la evaluación del algoritmo de la divergencia de Jensen-Shannon.

Figura 13. Vectores de distancia para el caso de inconsistencias en el micrófono, mediante divergencia de Jensen-Shannon



3.4. DETECCIÓN DE INCONSISTENCIAS DE MICRÓFONO

Para evaluar la detección de inconsistencias de micrófono se utilizó la herramienta *Classification Learner* de MATLAB y se hallaron tres clasificadores para cada distancia analizada, utilizando validación cruzada con 5 particiones y los datos de la fase de evaluación. Las características de los clasificadores utilizados se presentan a continuación:

- **SVM**: Máquina de vectores de soporte.
- **ANN**: Red neuronal artificial con una capa oculta de 10 nodos.
- **k-NN**: k-vecinos más cercanos con 10 vecinos.

Los parámetros de entrada para los clasificadores fueron:

- *Dif (d)*: Diferencia entre el valor máximo y mínimo de distancia

- $\varepsilon(d)$: Entropia wavelet de Shannon normalizada. Este parámetro se halló utilizando la función *wentropy* de MATLAB, concretamente se utilizó el tercer elemento de la entropia de nivel 2, hallada utilizando el método modwt (*maximal overlap discrete wavelet transform*) y la ondícula $sym4$.

Para el caso de los métodos LMNN, NCA e ITML de la distancia de Mahalanobis, los clasificadores cuentan con 2 entradas, mientras que en el caso en el que se evaluaron los tres métodos juntos, el número de entradas es 6.

En la tabla 2 se encuentran los resultados de la distancia de Mahalanobis, tanto de forma individual para cada método, como en forma conjunta para los tres métodos.

Tabla 2. Resultados de la Distancia de Mahalanobis

Clasificador	Método	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
SVM	LMNN	77.7	82.6	17.4	22.3	0.802	0.817	0.777	0.797
ANN	LMNN	77.6	81.9	18.1	22.4	0.798	0.811	0.776	0.793
k-NN	LMNN	74.6	82.5	17.5	25.4	0.786	0.810	0.746	0.777
SVM	NCA	75.4	80.9	19.1	24.6	0.782	0.798	0.754	0.775
ANN	NCA	77.6	79.1	20.9	22.4	0.784	0.788	0.776	0.782
k-NN	NCA	73	80.2	19.8	27	0.766	0.787	0.730	0.757
SVM	ITML	72.5	80.4	19.6	27.5	0.765	0.787	0.725	0.755
ANN	ITML	74.9	78.2	21.8	25.1	0.766	0.775	0.749	0.762
k-NN	ITML	69.5	80.3	19.7	30.5	0.749	0.779	0.695	0.735
SVM	Todos	77.8	89.8	10.2	22.2	0.838	0.884	0.778	0.828
ANN	Todos	79.3	87.7	12.3	20.7	0.835	0.866	0.793	0.828
k-NN	Todos	77.2	88.3	11.7	22.8	0.828	0.868	0.772	0.817

En la tabla 3 a continuación se presenta el promedio de los resultados de los tres clasificadores para cada método. A partir de los resultados de la tabla 3 se puede afirmar que el método que mejor desempeño individual tuvo fue LMNN; su precisión promedio fue 0.813, su recall promedio 0.766 y su valor F promedio 0.789.

También es posible afirmar que al evaluar los tres métodos juntos y, por consiguiente utilizando 6 entradas en los clasificadores, los resultados mejoraron considerablemente,

Tabla 3. Resultados promedio de la Distancia de Mahalanobis

Método	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
LMNN	76.6	82.3	17.7	23.4	0.795	0.813	0.766	0.789
NCA	75.3	80.1	19.9	24.7	0.777	0.791	0.753	0.771
ITML	72.3	79.6	20.4	27.7	0.760	0.780	0.723	0.750
Todos	78.1	88.6	11.4	21.9	0.834	0.873	0.781	0.824

presentándose una precisión promedio de 0.873, un *recall* de 0.781 y un valor F de 0.824. La buena mejora de la precisión promedio se debió, tal como se puede inferir a partir de las tablas 2 y 3, a la mejora de 3.37 por ciento en el promedio de los verdaderos positivos y a la disminución de 7.93 por ciento en el promedio de falsos positivos.

En la tabla 4 a continuación se presentan los resultados de la distancia probabilística hallada a partir de la divergencia de Jensen-Shannon y en la tabla 5, el promedio de estos resultados. A partir de los resultados de la tablas 4 y 5 se puede afirmar que el método que mejor desempeño tuvo para el caso de la distancia probabilística fue $p6$, es decir, aquel en el que se halló el valor máximo de divergencia de Jensen-Shanon entre aquellos correspondientes a los 6 primeros filtros del banco de filtros utilizado. De otro lado, los resultados generales del método $u30$ se vieron notoriamente afectados por una tasa de falsos negativos promedio de 37.3.

Tabla 4. Resultados de la Divergencia de Jenssen Shannon

Clasificador	Método	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
SVM	p6	79.6	93.0	7.0	20.4	0.863	0.919	0.796	0.853
ANN	p6	81.2	91.6	8.4	18.8	0.864	0.906	0.812	0.857
k-NN	p6	78.5	92.4	7.6	21.5	0.855	0.912	0.785	0.844
SVM	u30	62.1	87.8	12.2	37.9	0.750	0.836	0.621	0.713
ANN	u30	64.2	86.8	13.2	35.8	0.755	0.829	0.642	0.724
k-NN	u30	61.9	86.5	13.5	38.1	0.742	0.821	0.619	0.706
SVM	ambos	81.7	95.0	5.0	18.3	0.884	0.942	0.817	0.875
ANN	ambos	83.3	94.2	5.8	16.7	0.888	0.935	0.833	0.881
k-NN	ambos	81.9	95.3	4.7	18.1	0.886	0.946	0.819	0.878

Tabla 5. Resultados promedio de la Divergencia de Jensen-Shannon

Método	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
p6	79.8	92.3	7.7	20.2	0.861	0.912	0.798	0.851
u30	62.7	87.0	13.0	37.3	0.749	0.829	0.627	0.714
ambos	82.3	94.8	5.2	17.7	0.886	0.941	0.823	0.878

A pesar de unos resultados individuales no tan buenos del método $u30$, al unir las 4 características dadas por ambos métodos ($p6$ y $u30$) se obtuvieron mejores resultados en la clasificación, obteniéndose los siguientes resultados finales para el método de la Divergencia de Jensen-Shannon: precisión de 0.941, recall de 0.823 y valor F de 0.878.

En la tabla 6 a continuación se muestran los resultados finales de cada uno de los dos algoritmos implementados y evaluados en este trabajo. A partir de la tabla 6 se puede afirmar que el algoritmo del cual se obtuvieron mejores resultados fue el de la Divergencia de Jensen-Shannon.

Tabla 6. Resultados finales individuales de ambos algoritmos

	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
Mahalanobis	78.1	88.6	11.4	21.9	0.834	0.873	0.781	0.824
Jensen-Shannon	82.3	94.8	5.2	17.7	0.886	0.941	0.823	0.878

Los resultados del algoritmo que utiliza la Divergencia de Jensen-Shannon fueron mejores respecto a los del algoritmo que utiliza la Distancia de Mahalanobis según se lista a continuación:

- VP: mejora en 4.2%.
- VN: mejora en 6.2%.
- FP: disminuye en 6.2%.
- FN: disminuye en 4.2%.
- Exactitud: mejora en 5.2%.
- Precisión: mejora en 6.8%.

- Recall: mejora en 4.2 %.
- Valor F: mejora en 5.4 %.

Por último, en la tabla 7 se muestran los resultados en conjunto de los 5 métodos implementados en el presente trabajo: LMNNN, NCA, ITML, DJS-p6 y JSD-u30. Por lo tanto, los clasificadores listados en esta tabla fueron hallados con 10 parámetros de entrada.

Tabla 7. Resultados finales del conjunto de todos los algoritmos implementados

Clasificador	Método	VP	VN	FP	FN	Exactitud	Precisión	Recall	Valor F
SVM	DM y DJS	86.8	96.6	3.4	13.2	0.917	0.962	0.868	0.913
ANN	DM y DJS	88.7	95.3	4.7	11.3	0.920	0.950	0.887	0.917
k-NN	DM y DJS	84.9	96.7	3.3	15.1	0.908	0.963	0.849	0.902
Promedio	DM y DJS	86.8	96.2	3.8	13.2	0.915	0.958	0.868	0.911

En la tabla 7 se muestra que los resultados finales de clasificación, obtenidos al unir cada par de parámetros de los 5 métodos implementados en el presente trabajo, son los siguientes: exactitud: 0.915, precisión: 0.958, recall: 0.868 y valor F: 0.911.

De lo anterior se puede afirmar que tras considerar todos los métodos juntos, se obtuvo un resultado mucho mejor que con cada método individual, llegando a obtener unas tasas promedio de falsos negativos de 13.2 %, la cual es mejor a las que se obtuvieron individualmente con cualquiera de los métodos implementados.

4. CONCLUSIONES

Se implementaron y evaluaron de manera exitosa, tanto de forma individual como conjunta, dos algoritmos basados en el aprendizaje de métricas de distancia para la detección de inconsistencias de micrófono en grabaciones de audio. Se obtuvieron buenos resultados en cuanto al valor F (superior al $90\,\%$).

Se implementaron de manera satisfactoria tres métodos distintos de entrenamiento de distancias de Mahalanobis, obteniéndose valores de 0.781 y 0.824 para las medidas de *recall* y valor F, respectivamente. Se encontró que estas distancias enfatizan (dan mayor relevancia) a los rangos de frecuencias menores a $100\ Hz$ y superiores a $4\ kHz$, aproximadamente.

De otra parte, se implementó la divergencia de Jensen-Shannon para aquellos filtros triangulares ubicados en rangos de frecuencias menores a $103\ Hz$ y superiores a $4048\ Hz$. Esta distancia entregó mejores resultados que utilizar la distancia de Mahalanobis, obteniéndose valores de 0.823 y 0.878 para las medidas de recall y valor F, respectivamente.

Finalmente, se logró analizar el resultado del conjunto de todas las distancias implementadas. Este conjunto entregó una buena mejoría en comparación a distancias de Mahalanobis y probabilísticas por separado. Estos resultados finales fueron una exactitud de 0.915, precisión de 0.958, *recall* de 0.868 y valor F de 0.911.

5. RECOMENDACIONES PARA TRABAJOS FUTUROS

Recientemente se ha mostrado interés por el desarrollo de métricas basadas en *deep learning*, con las cuales se pueden obtener distancias métricas de mayor complejidad. Con las distancias de Mahalanobis se obtuvieron buenos resultados a pesar de corresponder a una transformación de tipo lineal en su espacio original. Se espera poder llegar a obtener muy buenos resultados al utilizar distancias métricas basadas en *deep learning*.

De otra parte, las distancias implementadas son solo un par de opciones dentro de un conjunto mayor de posibilidades. Otra pregunta que queda por resolver es la de analizar el desempeño utilizando otros tipos de distancias métricas, incluyendo las del tipo que no requieren entrenamiento de las mismas.

BIBLIOGRAFÍA

Brookes, Mike. *VOICEBOX: Speech Processing Toolbox for MATLAB*. http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html (vid. págs. 20, 35).

Capoferri, Davide et al. "Speech Audio Splicing Detection and Localization Exploiting Reverberation Cues". En: *2020 IEEE International Workshop on Information Forensics and Security (WIFS)* (2020), págs. 1-6 (vid. pág. 13).

Colombia, Fiscalia General de la Nación. República de. *Manual Único de Criminalística* (vid. pág. 12).

Davis, Jason V. et al. "Information-Theoretic Metric Learning". En: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, 209–216. DOI: 10.1145/1273496.1273523 (vid. págs. 24, 27).

de Vazelhes, William et al. "metric-learn: Metric Learning Algorithms in Python". En: *Journal of Machine Learning Research* 21.138 (2020), págs. 1-6 (vid. pág. 35).

Deza, Michel Marie y Elena Deza. *Encyclopedia of Distances*. 2nd ed. Springer, 2013. DOI: https://doi.org/10.1007/978-3-642-30958-8 (vid. págs. 14, 22).

Ergünay, S. K. et al. "On the vulnerability of speaker verification to realistic voice spoofing". En: *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. Sep. de 2015, págs. 1-6. DOI: 10.1109/BTAS.2015.7358783 (vid. págs. 32-34).

Esquef, P. A. A., J. A. Apolinário y L. W. P. Biscainho. “Edit Detection in Speech Recordings via Instantaneous Electric Network Frequency Variations”. En: *IEEE Transactions on Information Forensics and Security* 9.12 (dic. de 2014), págs. 2314-2326 (vid. pág. 13).

Garcia-Romero, Daniel y Carol Y. Espy-Wilson. “Automatic acquisition device identification from speech recordings”. En: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, págs. 1806-1809. DOI: 10.1109/ICASSP.2010.5495407 (vid. pág. 19).

Goldberger, Jacob et al. “Neighbourhood Components Analysis”. En: *Advances in Neural Information Processing Systems*. Ed. por L. Saul, Y. Weiss y L. Bottou. Vol. 17. MIT Press, 2004 (vid. pág. 23).

Gupta, S., S. Cho y C. C. J. Kuo. “Current Developments and Future Trends in Audio Authentication”. En: *IEEE MultiMedia* 19.1 (ene. de 2012), págs. 50-59. DOI: 10.1109/MMUL.2011.74 (vid. pág. 13).

Mcglynn, Matt. *RecordingHacks, a microphone database and search engine*. http://recordinghacks.com/ (vid. pág. 37).

Romito, L. y V. Galatà. “Towards a protocol in speaker recognition analysis”. En: *Forensic Science International* 146, Supplement (2004). Mediterranean Academy of Forensic Sciences 1st Workshop, S107 -S111. DOI: http://dx.doi.org/10.1016/j.forsciint.2004.09.033 (vid. pág. 12).

Suárez, Juan Luis, Salvador García y Francisco Herrera. “A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges”. En: *Neurocomputing* 425 (2021), págs. 300-322. DOI: https://doi.org/10.1016/j.neucom.2020.08.017 (vid. pág. 14).

Weinberger, Kilian Q, John Blitzer y Lawrence K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. En: *Advances in Neural Information Processing Systems 18*. Ed. por Y. Weiss, B. Schölkopf y J. C. Platt. MIT Press, 2006, págs. 1473-1480 (vid. págs. 23, 25).

Wu, Zhizheng et al. “SAS: A speaker verification spoofing database containing diverse attacks”. En: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, págs. 4440-4444. DOI: 10.1109/ICASSP.2015.7178810 (vid. pág. 33).

Zhao, Hong et al. “Audio splicing detection and localization using environmental signature”. En: *Multimedia Tools and Applications* 76 (2017), 13897–13927. DOI: 10.1007/s11042-016-3758-7 (vid. pág. 13).

ANEXOS

Anexo A. Banco de Filtros de mel

En la tabla a continuación se presentan los límites en frecuencia de los filtros de mel utilizados en el presente trabajo.

Tabla 8. Banco de filtros en escala mel

Filtro	f [Hz]	Filtro	f [Hz]	Filtro	f [Hz]	Filtro	f [Hz]	Filtro	f [Hz]
1	[0, 28]	27	[463, 510]	53	[1233, 1310]	79	[2512, 2640]	105	[4639 4851]
2	[14, 42]	28	[486, 533]	54	[1271, 1350]	80	[2576 2706]	106	[4744, 4961]
3	[28, 57]	29	[510, 558]	55	[1310, 1390]	81	[2640 2774]	107	[4851, 5072]
4	[42, 72]	30	[533, 583]	56	[1350, 1431]	82	[2706 2842]	108	[4961, 5186]
5	[57, 87]	31	[558, 608]	57	[1390, 1473]	83	[2774 2912]	109	[5072, 5302]
6	[72, 103]	32	[583, 634]	58	[1431, 1516]	84	[2842 2983]	110	[5186, 5421]
7	[87, 118]	33	[608, 660]	59	[1473, 1560]	85	[2912 3056]	111	[5302, 5542]
8	[103, 135]	34	[634, 687]	60	[1516, 1605]	86	[2983, 3130]	112	[5421, 5665]
9	[118, 151]	35	[660, 714]	61	[1560, 1650]	87	[3056, 3206]	113	[5542, 5790]
10	[135, 168]	36	[687, 742]	62	[1605, 1697]	88	[3130, 3283]	114	[5665, 5918]
11	[151, 185]	37	[714, 771]	63	[1650, 1744]	89	[3206, 3361]	115	[5790, 6049]
12	[168, 202]	38	[742, 800]	64	[1697, 1792]	90	[3283, 3441]	116	[5918, 6182]
13	[185, 220]	39	[771, 829]	65	[1744, 1841]	91	[3361, 3523]	117	[6049, 6318]
14	[202, 238]	40	[800, 859]	66	[1792, 1891]	92	[3441, 3606]	118	[6182, 6456]
15	[220, 257]	41	[829, 890]	67	[1841, 1942]	93	[3523, 3691]	119	[6318, 6597]
16	[238, 276]	42	[859, 921]	68	[1891, 1995]	94	[3606, 3778]	120	[6456, 6741]
17	[257, 295]	43	[890, 953]	69	[1942, 2048]	95	[3691, 3866]	121	[6597, 6888]
18	[276, 315]	44	[921, 986]	70	[1995, 2102]	96	[3778, 3956]	122	[6741, 7038]
19	[295, 335]	45	[953, 1019]	71	[2048, 2157]	97	[3866, 4048]	123	[6888, 7190]
20	[315, 355]	46	[986, 1053]	72	[2102, 2214]	98	[3956, 4142]	124	[7038, 7346]
21	[335, 376]	47	[1019, 1088]	73	[2157, 2271]	99	[4048, 4237]	125	[7190, 7505]
22	[355, 397]	48	[1053, 1123]	74	[2214, 2330]	100	[4142, 4335]	126	[7346, 7667]
23	[376, 419]	49	[1088, 1159]	75	[2271, 2389]	101	[4237, 4434]	127	[7505, 7832]
24	[397, 441]	50	[1123, 1196]	76	[2330, 2450]	102	[4335, 4535]	128	[7667, 8000]
25	[419, 463]	51	[1159, 1233]	77	[2389, 2512]	103	[4434, 4639]		
26	[441, 486]	52	[1196, 1271]	78	[2450, 2576]	104	[4535, 4744]		