Reconocimiento de comandos de voz mediante refuerzo de gradiente basado en histograma

Voice command recognition using histogram-based gradient boosting (HGB)

Teth Azrael Cortes Aguilar*, Adriana Tovar Arriaga

Departamento de Sistemas Computacionales y Electrónica
Tecnológico Nacional de México. Instituto Tecnológico José Mario Molina Pasquel y Henríquez
Camino Arenero 1101, Col. El Bajío, CP. 45017. Zapopan, Jalisco, México
*teth.cortes@zapopan.tecmm.edu.mx

PALABRAS CLAVE:

Reconocimiento de comandos de voz, extracción de características, refuerzo de gradiente

RESUMEN:

En el método tradicional de reconocimiento automático de voz, las señales de voz se traducen a texto para identificar palabras. Sin embargo, en algunos casos, como la robótica colaborativa, se requiere un reconocimiento rápido de comandos de voz, evitando la transformación de fonemas en texto. La extracción de características de las señales de voz es un paso crítico de preprocesamiento antes del entrenamiento de un modelo de aprendizaje automático. En este artículo se describe el método de extracción de características MFCC, CHROMA, TZ en el dominio de la frecuencia y RMS, ZC, TEMPO en el dominio del tiempo. Se creó un conjunto de datos de 600 grabaciones de audio. Para cada uno de los 10 comandos en español se hicieron 30 grabaciones de 20 hablantes adultos. En el análisis de la importancia de las características, concluimos que las características CHROMA no proporcionan datos informativos que mejoren la tarea de clasificación de señales de voz. Los resultados muestran que el reconocimiento de comandos de voz mediante un modelo de refuerzo de gradiente basado en histogramas HGB logra predicciones correctas en un 80% de los casos de prueba en 21.77 ms. Este modelo se comparó contra otros modelos de aprendizaje automático, logrando una mejor métrica de exactitud.

KEYWORDS:

Voice command recognition, feature extraction, gradient boosting

ABSTRACT:

In traditional automatic speech recognition methods, voice signals are translated into text to identify words. However, in some instances such as collaborative robotics, it requires fast voice commands recognition, bypassing the transformation of phonemes into text. The feature extraction from voice signals is a critical pre-processing step before training a machine learning model. This paper describes the method of features extraction MFCC, CHROMA, TZ in the frequency domain and RMS, ZC, TEMPO in the time domain. A dataset of 600 audio recordings was created. For each of the 10 commands in Spanish, 30 recordings were made by 20 adult speakers. In the analysis of the feature importance, we conclude that CHROMA features do not provide informative data that improve the task of classifying the voice signals. The results show that recognition of voice commands using histogram-based gradient boosting HGB model achieves true predictions in 80% of test cases within 21.77 ms. This model was compared against other machine learning models, and achieved better accuracy metric.

• Recibido: 4 de julio de 2025 • Aceptado: 25 de marzo de 2026 • Publicado en línea: 2 de octubre de 2026

Copyright:© 2026 *por los autores*
Creative Commons Atribución 4.0 CC-BY

1. INTRODUCCIÓN

En la industria, el uso de sistemas de reconocimiento de comandos de voz puede mejorar la interacción entre las personas y las máquinas. A diferencia de la mayoría de los sistemas de reconocimiento automático del habla ASR (*Automatic Speech Recognition*) que se enfocan en la traducción de las señales de voz en un formato de texto para identificar palabras o un hablante. En el reconocimiento de comandos se busca reconocer frases cortas distintas entre sí, a partir de las características extraídas de las señales acústicas en el dominio del tiempo o de la frecuencia. El reconocimiento de comandos de voz tiene aplicaciones importantes, en la domótica y la accesibilidad a dispositivos tecnológicos, en especial, para personas con alguna discapacidad. También mejora la seguridad en la conducción de vehículos y en la manufactura se está utilizando recientemente para el control de robots colaborativos [1].

En el presente trabajo de investigación se construyó un conjunto de datos de 10 comandos de voz en español, enfocados al control de un manipulador robótico colaborativo. Los comandos se etiquetaron como: C1 Desplázate hacia arriba, C2 Desplázate hacia abajo, C3 Muévete hacia adelante, C4 Muévete hacia atrás, C5 Gira a tu derecha, C6 Gira a tu izquierda, C7 Abre la pinza, C8 Cierra la pinza, C9 Muévete al origen y C10 Para ahí. Se obtuvieron características de las señales de audio tanto en el dominio del tiempo como de la frecuencia. Del espectro de frecuencia se obtuvieron las características MFCC (*Mel-Frequency Cepstral Coefficients*), CHROMA y TZ (*Tonnetz*); del oscilograma en el dominio del tiempo se obtuvieron RMS (*Root-Mean Square*), ZC (*Zero Crossing rate*) y TEMPO. Para la extracción de las características espectrales se realizó previamente la transformada de Fourier de corto plazo STFT (*Short-Term Fourier Transform*) de las señales. En la Figura 1 se muestran tres ejemplos de oscilogramas y sus respectivos espectrogramas para tres comandos y hablantes distintos. Las características extraídas de cada señal en el dominio del tiempo y de la frecuencia se unieron en un vector de 111 elementos, que se usaron para entrenar un algoritmo de aprendizaje supervisado de refuerzo de gradiente HGB (*Histogram-Based Gradient Boosting*) para realizar la tarea de reconocer y clasificar las señales según la clase del comando de voz.

El contenido del presente artículo está dividido de la siguiente forma. En la sección 2 se describe el algoritmo de refuerzo por gradiente. En la sección 3 se describen algunos trabajos relacionados. En la sección 4 se describe el método empleado para la extracción de las características de las señales de audio, también se describe el dataset y los parámetros de sintonización del modelo de refuerzo de gradiente basado en histograma. En la sección 5 de resultados se presenta una comparación de las métricas de rendimiento de once modelos de aprendizaje. En la sección 6 se expone la discusión de los resultados. Finalmente, en la sección 7 se presentan las conclusiones.

2. REFUERZO POR GRADIENTE

El refuerzo de gradiente basado en histograma HGB de la librería Scikit-Learn [2] es un algoritmo de clasificación de aprendizaje automático (*Machine Learning*) sustentado en modelos adaptativos, utilizado para entrenar modelos de aprendizaje débiles como árboles de decisión, más rápido que otros algoritmos de refuerzo de gradiente como AdaBoost [3] y XGBoost [4]. El modelo HGB fue agregado a Scikit-Learn en la versión 0.21 y está inspirado en el algoritmo LightGBM [5].

Un modelo de aprendizaje débil es un modelo tradicional que aprende de los datos de entrenamiento de forma independiente. Por otro lado, el refuerzo es una técnica de conjunto en el aprendizaje automático, que

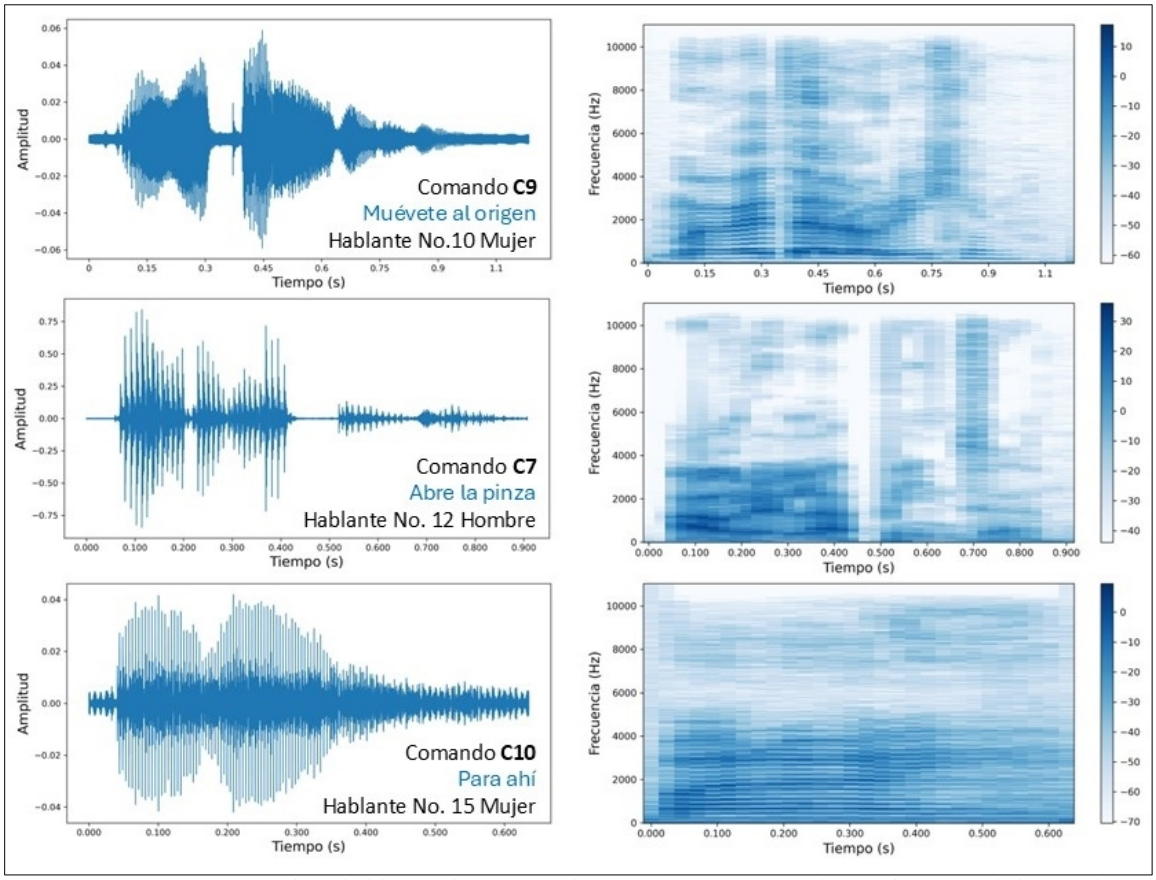

Figura 1. Para tres comandos y hablantes distintos, en la izquierda se muestran los oscilogramas de las señales acústicas y en la derecha se muestran sus respectivos espectrogramas. Fuente propia.

combina las predicciones de múltiples modelos débiles para crear un único modelo fuerte y más preciso. Los algoritmos de refuerzo de gradiente tienen un buen rendimiento para tareas de clasificación de aprendizaje supervisado con datos tabulados, y a diferencia de otros modelos, como redes neuronales que se comportan como cajas negras, son fáciles de interpretar.

En los algoritmos de refuerzo, primero se ajusta un modelo de aprendizaje débil, como un árbol de decisión, y se calculan los pseudoresiduos a partir de la diferencia entre el valor observado y el valor predicho. Después se construye otro árbol de decisión que predice los residuos; y se ajusta nuevamente el modelo de aprendizaje, con el propósito de reducir los nuevos pseudoresiduos, lo que significa que se está reduciendo la pérdida del modelo. Para obtener un modelo generalizado que haga predicciones correctas para datos nuevos y se evite el problema del sobreajuste (*overfitting*), los algoritmos de refuerzo de gradiente tienen un parámetro llamado tasa de aprendizaje, que escala la predicción de cada modelo débil y se define como un factor de multiplicación entre 0 y 1. El proceso anterior se repite para nuevos modelos de aprendizaje débil que agregan los residuos de la predicción inicial a la siguiente. Si después de cierto número de iteraciones no se mejora la pérdida, entonces se detiene el entrenamiento.La tasa de aprendizaje y el número de iteraciones son dos de los hiperparámetros del modelo HGB que se ajustan durante los entrenamientos. Tasas de aprendizaje con valores pequeños que tienden a cero, disminuyen el peso de cada modelo débil, lo cual eleva el tiempo de entrenamiento porque se necesitan construir

más árboles. No obstante, conforme se construyen más árboles, mejor rendimiento tendrá el modelo HGB, pero aumenta su complejidad y la posibilidad de sobreajuste, por lo que se suele emplear una combinación de baja tasa de aprendizaje y detención temprana del entrenamiento. La profundidad es otro hiperparámetro que controla el número de niveles de cada árbol de decisión. Se recomienda un valor mínimo de 3 para evitar el sobreajuste y un valor máximo de 10 para no elevar el costo computacional debido al incremento en la complejidad del algoritmo. Otro hiperparámetro es el número mínimo de muestras por hoja que determina en cuantas ramas se dividen los árboles de decisión. Un valor bajo de muestras incrementa la sensibilidad del algoritmo al ruido y un valor alto evita el sobreajuste [5].

La inducción en los árboles de decisión con un enfoque descendente funciona como un algoritmo *greedy*, donde en cada nodo del árbol se elige un par (f, f_T). Si la característica es menor que un valor umbral $f<f_T$ se continua con el siguiente nodo. En este caso, la toma de decisiones es de corto alcance y está en función de la información inmediata disponible sin considerar las consecuencias futuras. El objetivo es encontrar el menor par (f, f_T) de acuerdo a una métrica de entropía, como la ganancia de información o la impureza de Gini. En la ecuación (1) la entropía mide la impureza de los valores de una muestra que pueden estar entre 0 y 1. Donde S representa el conjunto de datos sobre el que se calcula la entropía. C representa las clases en el conjunto S y p_i es una fracción que representa la probabilidad de seleccionar aleatoriamente un dato que pertenece a la clase i. Como el resultado del logaritmo de una fracción es un valor negativo, la ecuación se multiplica por -1. Si todas las muestras de S pertenecen a una clase, entonces la entropía será igual a 0. Si la mitad de las muestras se clasifican en una clase y la otra mitad en otra clase, la entropía será igual a 1. La ganancia de información es la diferencia de entropía antes y después de una división de una característica determinada. La característica con la mayor ganancia de información producirá la mejor división. La impureza de Gini es la probabilidad de clasificar erróneamente un dato elegido aleatoriamente del conjunto de datos. Cuando mayor sea el valor del índice de Gini mayor será la probabilidad de clasificaciones incorrectas [6].

$$E(S) = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{1}$$

Para cada característica f y para cada valor umbral f_T se calcula la métrica de la ganancia de información o índice de Gini, hasta encontrar el par (f, f_T) con la mejor métrica. En cada nodo del algoritmo de refuerzo, la métrica de ganancia depende únicamente de la suma de los gradientes del hijo izquierdo, el hijo derecho y el nodo actual. Para calcular estas sumas de gradientes rápidamente, un enfoque típico consiste en ordenar los valores de la característica f que se está probando. Para cualquier algoritmo de ordenamiento basado en comparaciones la complejidad superior se alcanza en $O(|f| \cdot n \log n)$, donde $|f|$ es el número de características que se evalúan y n el número de valores a evaluar en cada característica [5], [6].

Como se muestra en la Figura 2, el algoritmo HGB crea un histograma de valores de características f, con igual densidad de intervalos (n_{bins}), es decir, transforma la distribución de características en una distribución uniforme. Este enfoque permite que el algoritmo aproveche estructuras de datos basadas en números enteros o histogramas en lugar de depender de valores continuos ordenados para construir los árboles. La suma de gradientes se almacena en cada intervalo del histograma, lo que permite un cálculo más rápido de la ganancia porque la complejidad del algoritmo se reduce a $O(n_{bins})$. En HGB el número de intervalos es $n_{bins}=255$. Además, la ganancia del nodo del hijo izquierdo y derecho se pueden

usar para calcular la ganancia de cada nodo padre. Esto significa la posibilidad de calcular la ganancia de un nodo hijo restando la ganancia del nodo padre, que fue calculado anteriormente y en la práctica se simplifica el proceso de construcción del histograma. Durante el entrenamiento de HGB, el generador de árboles aprende en cada punto de división y asigna los valores faltantes al hijo izquierdo o derecho, según la ganancia del nodo padre. Si no se encuentran valores faltantes para una característica dada entonces las muestras con valores faltantes se asignan al hijo con más muestras. En tareas de clasificación multiclase, HGB utiliza una función de pérdida logarítmica también denominada como entropía cruzada. Internamente, el modelo ajusta para cada clase un árbol por iteración de refuerzo, y utiliza la función softmax para calcular las probabilidades predichas de cada clase [2], [7].

3. TRABAJOS RELACIONADOS

En investigaciones recientes sobre aplicaciones distintas, se ha encontrado que modelos de aprendizaje por refuerzo de gradiente como LightGBM en el cual se basa el modelo HGB de Scikit-Learn, utilizado en el presente trabajo, se logran métricas de rendimiento mejores que otros modelos de aprendizaje. En [8] reportan que un algoritmo de refuerzo de gradiente logró mejores métricas de accuracy que modelos como SVM (*Support Vector Machine*), RFC (*Random Forest Classifier*), KNN (*k-Nearest Neighbors*) y NB (*Naive Bayes*) en el diagnóstico de diabetes mellitus. En [9] después de comparar ocho modelos de aprendizaje, concluyeron que el algoritmo de refuerzo de gradiente LightGBM, logró mejores métricas para detectar complicaciones comunes durante el embarazo, después de entrenar un modelo con datos de características obtenidas de muestras de la sangre de 149 pacientes con hipertensión y 250 colestasis intrahepática. En [10] reportan que LightGBM logró métricas de rendimiento mejores que SVM para reconocer emociones en mensajes de texto. En [11] los investigadores demostraron que el modelo de refuerzo de gradiente logró un mejor rendimiento para el diagnóstico de fallas en paneles fotovoltaicos que otros modelos como SVM, MLP, DTC y KNN. En [12] utilizaron un algoritmo de refuerzo por gradiente para detectar noticias falsas a partir del título de la reseña de un texto, los investigadores mencionan que su algoritmo es más exacto que otros modelos de aprendizaje como MNB (*Multinomial Naive Bayes*), LR (*linear regression*), SVM y LSTM (*Long short-term memory*), utilizaron cuatro métodos de vectorización para extraer características y reconocer palabras de mensajes de texto, para el mejor de los métodos de vectorización y usando LightGBM reportan una métrica de accuracy del 62.06%, superior al 58.87% de XGBoost. En [13] analizaron señales acústicas, a partir de la extracción de características en el dominio del tiempo y de la frecuencia, para diagnosticar fallas por sujeción de pernos, con LightGBM reportaron un mejor rendimiento de 5.47% y 2.21% mayor a XGboost, antes y después de la eliminación del ruido. En [14] utilizaron con una tasa de éxito del 95.26% el algoritmo LightGBM para reconocer el género del hablante, extrayendo características de señales de voz a partir del reconocimiento de patrones mediante FRT (*Forward Rajan Transform*). En otra investigación [15], fusionaron características espectrales de señales de audio, y reportaron que el algoritmo LightGBM logró la mejor métrica de exactitud en el reconocimiento del hablante. Para la clasificación de emociones a partir de señales de voz, en [16] usaron características del espectro de Mel, MFCC, CHROMA, transformada wavelet, curtosis, RMS y ZC para entrenar un modelo LightGBM, que logró una exactitud del 62.94% al 84.91% para conjuntos de datos distintos. Concatenando características espectrales de señales de acústicas, en [17] y [18] también reportan que el algoritmo LightGBM alcanzó una alta exactitud en la clasificación de palabras. Además de alcanzar

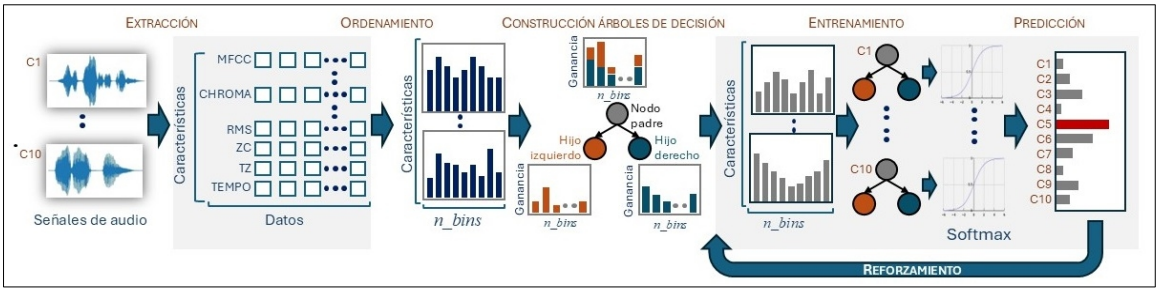

Figura 2. Extracción de características de las señales de audio, ordenamiento de las características en histogramas, construcción de árboles de decisión y reforzamiento en el modelo HGB. Fuente propia.

métricas altas de exactitud, en [19] se reportó que el algoritmo de refuerzo de gradiente basado en histograma también logró tiempos más cortos en predicciones que otros algoritmos de refuerzo de gradiente.

4. METODOLOGÍA

La extracción de características se realizó mediante la biblioteca librosa [20] de Python usando un equipo de cómputo ASUS TUF F15 con CPU INTEL i7, 16 GB en RAM y GPU NVIDIA RTX4050. El entrenamiento y prueba del modelo HGB se realizó en el mismo equipo de cómputo.

Para la extracción de las características MFCC y CHROMA se requiere obtener previamente el espectro de la señal de audio mediante la transformada de Fourier de corto plazo STFT. A diferencia de la transformada discreta de Fourier, que se calcula sumando la función en el tiempo, STFT considera solo un segmento o marco de tiempo de corta duración de una señal más larga y calcula su transformada de Fourier. En (2) el número complejo $X(m,k)$ denota el k-ésimo coeficiente de Fourier para el n-ésimo marco de tiempo fijo en m muestras. La señal de tiempo discreto denotada como $x(n+mH)$ se multiplica por una ventana de muestreo rectangular $w(n)$ con valores de $n=0,1,\ldots,N-1$. El parámetro de salto H, especifica la cantidad de muestras que se desplazará la ventana $w(n)$ a través de la señal de tiempo discreto. En el presente proyecto se estableció una longitud de ventana de 1024 muestras y un valor de 512 de salto.

$$y(m,k)=\left|X(m,k)\right|^2=\left|\sum_{n=0}^{N-1} x(n+mH)\,w(n)\,e^{-i2\pi\frac{k}{N}n}\right|^2 \quad (2)$$

Se estableció en 40 el número de coeficientes cepstrales y de cada uno se calculó la media y la desviación estándar, para obtener un total 80 características de MFCC del espectro de la señal de audio. Los coeficientes cepstrales se calculan a partir de STFT, mapeando el espectro resultante a la escala mel, calculando el logaritmo de las potencias en cada frecuencia mel y aplicando la transformada discreta del coseno. Cabe mencionar que la escala de mel se relaciona con la percepción auditiva humana, lo que permite la detección de voz, la identificación del hablante y reconocimiento de emociones.

Para CHROMA se estableció en 12 el parámetro de intervalos del espectro calculado a partir de STFT. Para cada intervalo se calculó la media y la desviación estándar, para obtener un total de 24 características. La escala cromática se compone de octavas que abarcan 12 sonidos denominados semitonos. Una octava se define como un intervalo de frecuencias entre dos sonidos cuya frecuencia central es del doble para intervalos sucesivos. Es decir, si la frecuencia de la nota musical Do de la primera octava es 32.7032 Hz, la de la segunda octava será 65.4064Hz y así sucesivamente. La escala cromática se suele usar para identificar el tipo de instrumento a partir del sonido que produce, así como

reconocer el tipo de composición musical [21].

Las características de la media y la desviación estándar de TZ se utilizan para detectar cambios en el contenido armónico del espectro de la señal que dependen de la anatomía vocal del hablante, su interpretación, su intensidad y su forma de expulsar el aire al hablar. Se suele usar para distinguir una voz femenina fuerte y clara, de una voz masculina grave y suave [22]. Cabe precisar que en el conjunto de datos usado para el entrenamiento del modelo HGB, el 65% de las grabaciones de audio corresponden al género masculino y el 35% al género femenino.

El valor cuadrático medio RMS se puede calcular a partir de las muestras del oscilograma o del espectrograma. Aunque es más rápido obtener el valor RMS a partir de muestras de la señal de audio en el dominio del tiempo, porque no se necesita calcular STFT, es preferible obtener el valor RMS a partir de un espectrograma porque brinda una representación más precisa de la energía a lo largo del tiempo, debido a que el espectro de frecuencia se divide en ventanas de muestreo más cortas. Las características extraídas del valor RMS son la media y la desviación estándar.

La tasa de cruce por cero ZC representa la cantidad de veces que la señal cambia de signo, de positivo a negativo y viceversa, dividida por la longitud del segmento de audio. La tasa de cruce por cero se suele utilizar para la detección de tono y determinar la presencia de voz humana, también se suele utilizar para clasificar sonidos de percusión [23]. Las características extraídas de ZC son la media y la desviación estándar.

De la función de TEMPO solo se extrae una característica relacionada con el número de pulsaciones por minuto de la señal, que se asocia con la rapidez a la que se pronuncia un comando de voz. El ritmo o TEMPO se obtiene de la envolvente de la señal de audio en el dominio del tiempo y muestra los cambios más significativos de su amplitud, sin importar el tono o la frecuencia.

4.1. Dataset

Dado que la detección de voz varía según el idioma, los algoritmos de reconocimiento son inherentemente sensibles a la fonética específica del idioma. En este contexto, se construyó un conjunto de datos para el presente proyecto, que está ordenado en 20 carpetas, una para cada hablante, ver Figura 3. Dentro de cada carpeta, hay 30 archivos de audio wav. Cada archivo se identificó con cuatro pares de números XX-XX-XX-XX.wav. El primer par designa al hablante del 01 al 20, el segundo par indica el género, 01 para masculino y 02 para femenino. El tercer par indica la emoción del hablante: 01 para enojo, 02 para neutro y 03 para tristeza. Finalmente, el cuarto par identifica el comando de voz en español utilizado para controlar un brazo robótico con una pinza como herramienta. Los comandos son: 01 Desplázate hacia arriba, 02 Desplázate hacia abajo, 03 Muévete hacia adelante, 04 Muévete hacia atrás, 05 Gira a tu derecha 06 Gira a tu izquierda, 07 Abre la pinza, 08 Cierra la pinza, 09 Muévete al origen y 10 Para ahí.

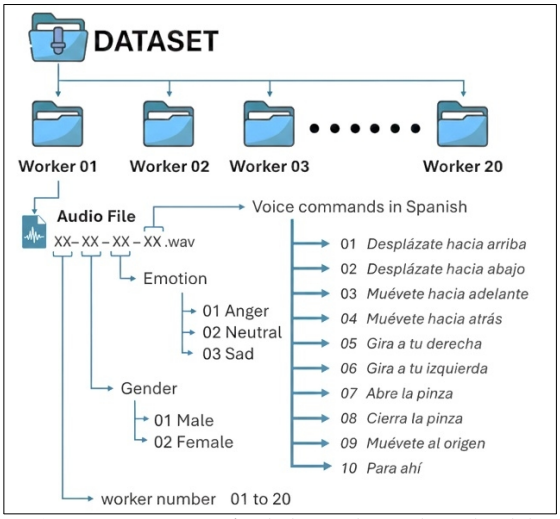

Figura 3. Organización de los archivos de audio del Dataset *Spanish voice command recognition*. Fuente [24].

El dataset consta de 600 grabaciones de audio; de cada comando de voz se hicieron 30 grabaciones de 20 hablantes adultos de entre 20 y 60 años de edad, de los cuales 13 eran hombres y 7 mujeres. Las grabaciones se realizaron con un micrófono unidireccional de condensador con filtro anti ruido, ajuste de volumen y control de la profundidad de la voz, modelo MIC-556, de la marca STEREN. Las grabaciones de cada comando de voz se guardaron en un formato digital wav sin compresión. Se estableció una frecuencia de muestreo de 22.050 kHz para la importación de los archivos de audio al entorno de programación; suponiendo una frecuencia de corte de 11.025 kHz como la frecuencia máxima audible. Los valores de la media y la desviación estándar de las características extraídas por MFCC, CHROMA, TZ, ZC y RMS se multiplicaron por un factor de 1000 para igualar su magnitud a los valores típicos de la característica TEMPO, antes de enviarlos al modelo HGB. En total se utilizaron 111 características, de las cuales 106 corresponden al dominio de la frecuencia y 5 al dominio del tiempo. Para la etapa de entrenamiento (*Train*) del modelo HGB se utilizaron 570 grabaciones y para la etapa de prueba (*Test*) se utilizaron 30 grabaciones seleccionadas aleatoriamente y que no se usaron previamente en la etapa de entrenamiento. En diez procedimientos la extracción de las características de los datos de entrenamiento se requirió una media de 52.2 s con una desviación estándar de 1.15 s.

4.2. Parámetros de sintonización

En la Tabla 1 se muestran los parámetros de sintonización del modelo HGB. Se usó una función de pérdida de entropía cruzada categórica Log_loss en el proceso de refuerzo. El mejor factor de multiplicación de las hojas, encontrado durante la sintonización del modelo HGB fue de 0.25. El número máximo de iteraciones se incrementó gradualmente durante el entrenamiento hasta llegar a 60, para tareas de clasificación con n número de clases se construyen n árboles por clase por iteración. El número máximo de hojas por árbol se estableció en 30. No se limitó la profundidad del árbol. No se modificó el número mínimo de muestras por hoja, dejando el parámetro en 20. No se utilizó un parámetro de regularización L2. El parámetro max_features que determina la proporción de características elegidas aleatoriamente en cada nodo, se estableció en un valor bajo de 1 para construir árboles de aprendizaje débil y prevenir el sobreajuste del modelo. El número máximo de intervalos del histograma se estableció en 255. Todas las características de entrada al modelo HGB se consideraron de tipo numérico.

Con el propósito de comparar el rendimiento del modelo HGB contra otros modelos de aprendizaje automático, también se entrenaron los modelos SVM (*Support Vector Machine*), DTC (*Decision Tree Classifier*), RFC (*Random Forest Classifier*), QDA (*Quadratic Discriminant Analysis*), GPC (*Gaussian Process Classifier*), ERT (*Extremely Randomized Tree classifier*), MLP (*Multi-Layer Perceptron*), GNB (*Gaussian Naive Bayes*), ADB (AdaBoost Classifier) y KNN (*k-Nearest Neighbors*). En la Tabla 2 se muestra un resumen de los principales parámetros de sintonización de estos modelos.

5. RESULTADOS

Con el objetivo de identificar cuáles son los datos informativos, en la Figura 4, se muestra la importancia de las características para la tarea de clasificación en los modelos de árboles de decisión DTC y bosques aleatorios RFC. Cabe recordar que el modelo HGB utiliza árboles de decisión como modelos de aprendizaje débil. Las primeras 80 barras corresponden a la importancia de las características extraídas por MFCC de la media μ y la desviación estándar σ de las componentes espectrales de las señales de audio. Las siguientes 24 barras corresponden a la importancia de las características extraídas mediante los bancos de filtros del espectro mediante la función CHROMA.

Tabla 1. Parámetros de sintonización del modelo HGB

Parámetro	Valor	Parámetro	Valor
loss	Log_loss	min_samples_leaf	20
learning_rate	0.25	l2_regularization	0
max_iter	60	max_features	1
max_leaf_nodes	30	max_bins	255
max_depth	None	categorical_features	None

Tabla 2. Principales parámetros de sintonización de otros modelos de aprendizaje

Modelo	Parámetros
SVM	kernel="linear", C=0.025, random_state=50, penalty='l2', loss='squared_hinge', max_iter=1000
DTC	criterion= 'entropy', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1
RFC	max_depth=10, n_estimators=23, max_features = 'sqrt', random_state=42, criterion ='entropy'
QDA	priors=None, reg_param=0.0, store_covariance=False, tol=1e-4
GPC	kernel=KRN, random_state=0, multi_class="one_vs_rest", max_iter_predict=100
ERT	BaggingClassifier, random_state=2, n_estimators=11
MLP	alpha=0.001, max_iter=1000, hidden_layer_sizes=100, activation='relu', solver='adam'
GNB	priors=None, var_smoothing=2e-10
ADB	learning_rate=1.45, random_state=50, estimator=None, n_estimators=50, algorithm='deprecated'
KNN	n_neighbors=4, algorithm='brute', weights='uniform', metric='minkowski', p=2

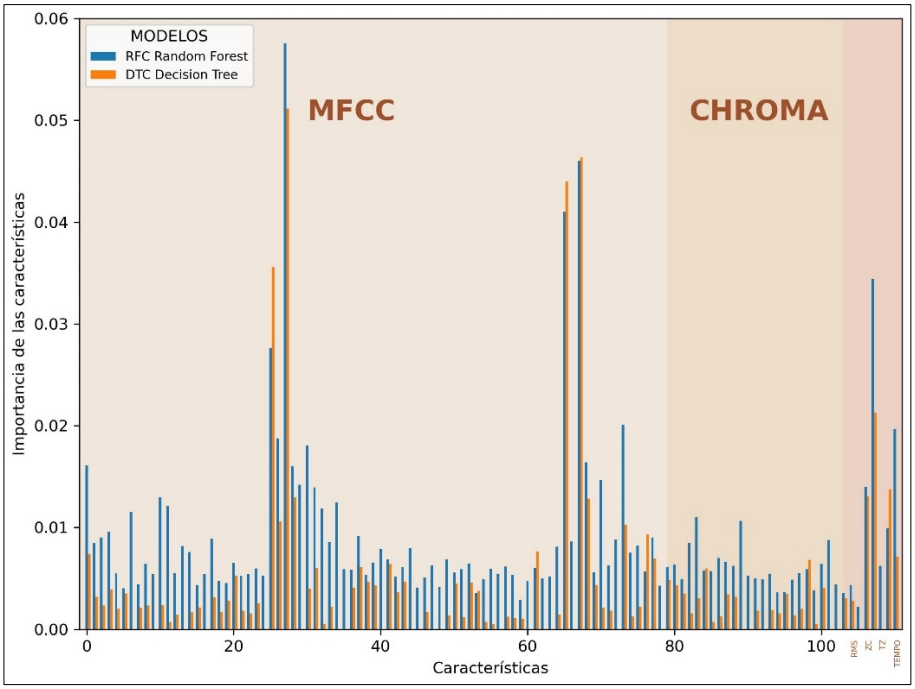

Figura 4. Importancia de las características en la tarea de clasificación en los modelos de árboles de decisión DTC y bosques aleatorios RFC. Fuente propia.

Después se muestra en las últimas 7 barras la importancia de las características de la señal en el dominio del tiempo, para la media μ y la desviación estándar σ de las funciones RMS, ZC, TZ, así como el número pulsaciones por minuto de la función TEMPO. Para la tarea de clasificación se observa que las características en el dominio de la frecuencia, extraídas a través de MFCC aportan información más relevante que las extraídas por CHROMA. También se observa que las características ZC, TZ y TEMPO son más relevantes que RMS.

Con una tasa de aprendizaje de 0.25 y un número máximo de hojas por nodo de 30. Se observa en la Figura 5 que el modelo HGB alcanza la máxima métrica de accuracy después de 10 iteraciones para los datos de entrenamiento. Considerando los datos

extraídos por CHROMA, después de 50 iteraciones la mejor métrica de accuracy para la curva de Test 1 es de 0.76, es decir, el modelo acertó en clasificar el 76% de las señales de audio con el comando de voz correcto. Entrenando nuevamente el modelo HGB, pero sin los datos de CHROMA, después de 35 iteraciones la mejor métrica de accuracy para la curva de Test 2 es de 0.8, es decir, se mejora un poco la exactitud de las predicciones del modelo. Este resultado está relacionado con la observación anterior de que las características extraídas por CHROMA aportan menor información relevante. Conviene señalar que la extracción de 111 características con los datos de CHROMA requiere aproximadamente un 5% más de tiempo que la extracción de 87 características, sin los datos de CHROMA.

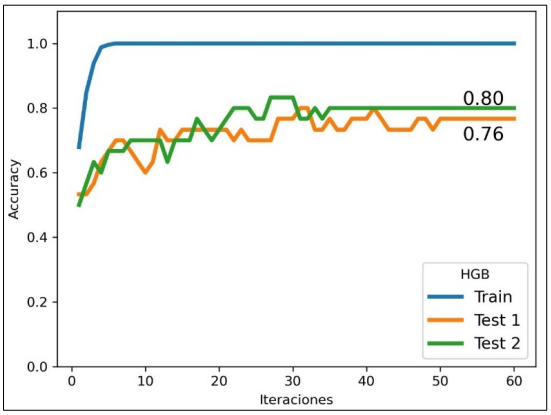

Figura 5. Comparación de la métrica accuracy, en Test 1 se entrenó el modelo con los datos de CHROMA y en Test 2 se entrenó el modelo sin los datos de CHROMA. Fuente propia.

En la imagen de la izquierda de la Figura 6 se observa que usando 570 datos de entrenamiento se logró una métrica de accuracy del 100%. Por otro lado, en la imagen de la derecha se observa que la matriz de confusión con el mejor resultado del modelo HGB, se logró con los parámetros de Test 2, reportando una métrica de accuracy del 80% para 30 datos, es decir, el modelo acertó en 24 ocasiones y falló en 6.

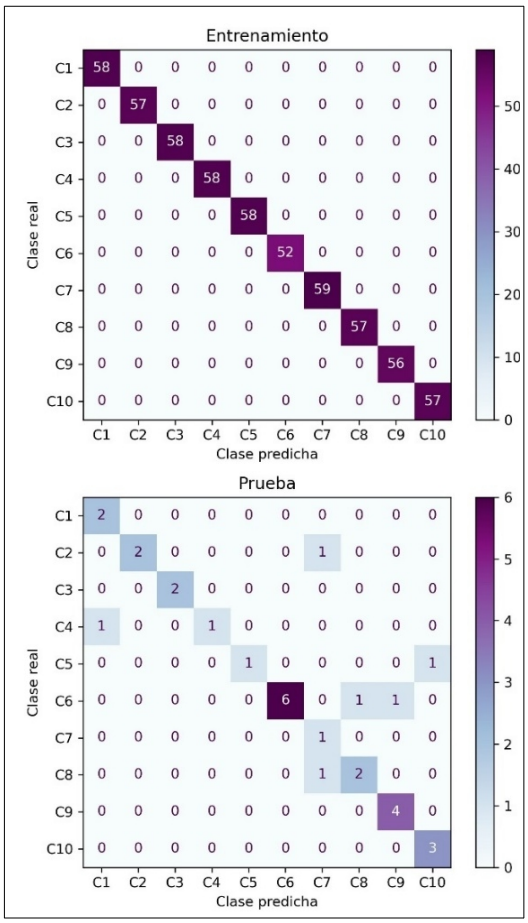

Figura 6. Matriz de confusión para los datos de entrenamiento y Test 2. Fuente propia.

Las 6 predicciones incorrectas del modelo HGB en Test 2, corresponden a los archivos del dataset: 10-02-03-06.wav, 11-01-02-02.wav, 12-01-02-06.wav, 14-01-02-05.wav, 14-01-03-04.wav y 15-02-01-08.wav. En particular, el modelo falló en las grabaciones de los hablantes 10, 11, 12, 14 y 15. En el hablante 10 un comando de voz C6 se predijo erróneamente como C9. En el hablante 11 un comando C2 se predijo de forma incorrecta como C7. En el hablante 12 un comando C6 se predijo equivocadamente como C8. En el hablante 14 se predijo erróneamente un comando C4 como C1 y un comando C5 como C10. Y en el hablante 15 el modelo clasificó erróneamente un comando C8 como C7. En la Tabla 3 se muestran los resultados de Test 2 de las métricas de *Precision*, *Recall* y *F1-score* para las predicciones del modelo HGB por cada clase o comando. La columna soporte

indica la cantidad de archivos utilizados en la prueba de clasificación. La métrica accuracy indica el porcentaje total de predicciones correctas, tanto positivas como negativas, entre el número total de datos de la prueba. La métrica *Precision* indica la proporción de predicciones positivas que fueron correctas, mientras que la métrica *Recall* mide la capacidad del modelo para identificar todos los casos positivos reales. F1-score es un promedio de las dos anteriores en una sola métrica.

Tabla 3. Resultados de las métricas de calidad de las predicciones del modelo de HGB

Clase	Precision	Recall	F1-score	Soporte
C1	0.67	1.00	0.80	2
C2	1.00	0.67	0.80	3
C3	1.00	1.00	1.00	2
C4	1.00	0.50	0.67	2
C5	1.00	0.50	0.67	2
C6	1.00	0.75	0.86	8
C7	0.33	1.00	0.50	1
C8	0.67	0.67	0.67	3
C9	0.80	1.00	0.89	4
C10	0.75	1.00	0.86	3
Accuracy			0.8	30

En la Figura 7 se muestra la evaluación del reconocimiento de comandos por hablante. Para todas las grabaciones de audio del dataset se volvieron a extraer las 87 características informativas, que se evaluaron con el método predict de la librería Scikit-Learn con los mismos parámetros del modelo HGB usados en Test 2. En la figura 7 se muestran en rojo las 6 predicciones incorrectas descritas anteriormente. Los hablantes 10 y 15 son mujeres mientras que los hablantes 11, 12 y 14 son hombres. Cabe señalar, que de cada comando se obtuvieron 3 grabaciones por hablante. Cada marca en color verde de la figura indica por lo tanto que el modelo acertó para las 3 grabaciones y el total de archivos evaluados corresponden a las 600 grabaciones del dataset.

En la gráfica de la Figura 8 se muestra una comparación de los resultados de la métrica accuracy para la tarea de clasificación de distintos modelos de aprendizaje automático. Para los datos de entrenamiento, se observa que los modelos KNN, ADB, GNB y MLP no lograron una métrica de accuracy de 1 en comparación a los demás modelos de aprendizaje. A pesar de que los modelos ERT, GPC, QDA alcanzan una métrica alta en la etapa de entrenamiento, muestran una métrica baja, menor al 0.5, para los datos de prueba. Los modelos RFC y DTC muestran un rendimiento similar acertando correctamente en el 56 % de los datos de prueba. El modelo SVM superó a los modelos RFC y DTC haciendo una predicción correcta en el 66 % de los casos. No obstante, fue el modelo HGB el que logró la mejor métrica de accuracy, clasificando correctamente el comando de voz del 76% de las señales de prueba.

En la gráfica de la Figura 9 se comparan los tiempos de entrenamiento de los distintos modelos de aprendizaje. Para 570 grabaciones de comandos de voz, el peor resultado lo obtuvo el modelo GPC con 12,448 ms. Pero, muy por debajo de este tiempo están el modelo HGB con 565 ms, MLP con 363 ms, ADB con 358, RFC con 157 ms, DTC con 94 ms, SVM con 57 ms, KNN con 20 ms, QDA con 19 ms, ERT con 16 ms y GNB con 12 ms.

6. DISCUSIÓN

En diez pruebas del modelo HGB previamente entrenado y sin considerar el tiempo de extracción de las características de la señal de audio, HGB requirió una media de 12.2 ms con una desviación estándar de 4.2 ms. Y el tiempo promedio que se requiere para extraer las características de una grabación de audio como del comando C3, conformado por tres palabras es de 9.57 ms. Es decir, el tiempo total para hacer una predicción, considerando la extracción de las 111 características es de 21.77 ms. Este tiempo total es un buen resultado para un algoritmo de aprendizaje automático si lo comparamos contra el tiempo de 130 ms que le toma a la corteza temporal izquierda del cerebro humano, distinguir oraciones simples de tres palabras [25]. Es decir, el modelo HGB es casi

Figura 7. Evaluación del reconocimiento de comandos por hablante.

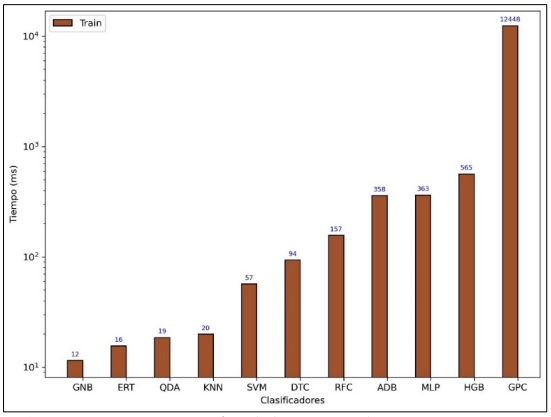

Figura 9. Comparación del tiempo de entrenamiento de los modelos de clasificación. Fuente propia.

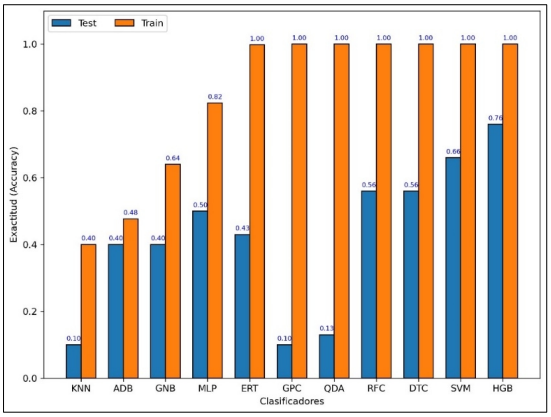

Figura 8. Comparación de la métrica accuracy para los datos de entrenamiento de los modelos de clasificación. Fuente propia.

seis veces más rápido que el cerebro humano para comprender el significado de una frase corta. Por otro lado, la métrica de accuracy del 76% al 80% del modelo HGB tiene un rendimiento similar al de otros métodos como ASR, basados en el reconocimiento de fonemas que también utilizan MFCC [26] para la extracción de características, por ejemplo, en [27] con un algoritmo de clasificación SVM lograron una métrica del 77.6%, en [28] con un algoritmo de clasificación HMM (*Hidden Markov Models*) lograron una métrica del 75.78% y en [29] con un modelo hibrido de HMM y MLP lograron una métrica de 77.83%, en estos casos los modelos de aprendizaje fueron entrenados con un conjunto de datos diferentes al idioma español. La construcción del dataset con comandos de voz en español, es otra contribución importante de este trabajo, sin embargo, como trabajo futuro se propone aumentar la cantidad de grabaciones de audio por comando de voz durante las pruebas del modelo HGB.

7. CONCLUSIONES

Se utilizó un conjunto de datos de 600 grabaciones de audio clasificados en diez comandos de voz en español, para entrenar un modelo de aprendizaje supervisado de

refuerzo de gradiente HGB. Se obtuvieron las características espectrales MFCC, CHROMA y TZ de las señales de audio mediante la transformada de Fourier de corto plazo STFT. También se obtuvieron las características RMS, ZC y TEMPO de las señales en el dominio del tiempo. Se utilizó un vector de entrada al modelo HGB de 111 datos, calculados a partir de la media y la desviación estándar de las características extraídas por MFCC, CHROMA, TZ, RMS, ZC y las pulsaciones por minuto de la función TEMPO. Se observó que las características extraídas por CHROMA aportan pocos datos informativos para la tarea de clasificación. Después de entrenar el modelo HGB, utilizando las características de CHROMA, en la etapa de prueba, el modelo HGB alcanzó una exactitud en las predicciones del 76%, después de 50 iteraciones y del 80% sin utilizar las características de CHROMA, después de 35 iteraciones. En comparación con otros modelos, HGB muestra una mejor métrica de rendimiento en la etapa de prueba usando los mismos datos que SVM, DTC, RFC, QDA, GPC, ERT, MLP, GNB, ADB y KNN. Además, el modelo HGB tiene un rendimiento similar o mejor al de otros métodos basados en el reconocimiento de fonemas que también utilizan MFCC para la extracción de características. El modelo HGB requiere un tiempo de entrenamiento más largo que otros modelos como SVM, DTC, RFC, QDA, ERT, MLP, GNB, ADB y KNN. Sin embargo, el tiempo que le toma extraer 111 características de una señal de audio y hacer una predicción es de 21.77 ms, que es adecuado para el reconocimiento de comandos de voz, por ejemplo, para el control de un manipulador robótico colaborativo.

Agradecimientos

Agradecemos al Ingeniero en electrónica Raúl Romero Valenzuela por su apoyo durante las grabaciones de los comandos de voz, así como a la U. A. Zapopan del Instituto Tecnológico José Mario Molina Pasquel y Henríquez ITJMMPyH, al Tecnológico Nacional de México TecNM y a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación SECIHTI del gobierno de México por el apoyo recibido para la divulgación del presente trabajo. .

REFERENCES

[1] Ayache M, Kanaan H, Kassir K, Kassir Y. Speech command recognition using deep learning. En: Sixth Int. Conf. on Advances in Biomedical Engineering (ICABME); 2021; Werdanyeh, Lebanon. p. 24-29. doi: 10.1109/ICABME53305.2021.9604862

[2] Pedregosa F, et al. Scikit-learn: machine learning in Python. Journal of Machine Learning Research (2011) 12(85):2825-2830. https://scikit-learn.org/stable/

[3] Schapire RE. Explaining AdaBoost. En: Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik. Springer Berlin Heidelberg (2013) p. 37-52. doi: 10.1007/978-3-642-41136-6_5

[4] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. En: Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '16); 2016; New York, NY, USA. Association for Computing Machinery. p. 785-794. doi: 10.1145/2939672.2939785

[5] Ke G, et al. LightGBM: A highly efficient gradient boosting decision tree. En: Proc. of the 31st Conf. on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4-9; Long Beach, USA. p. 3149-3157.

[6] Chugani V. ¿Qué es el refuerzo? Tutoriales de Ciencia de Datos. DataCamp. 11 Agosto 2024. [citado 4 de Marzo de 2024]. https://www.datacamp.com/es/tutorial/what-is-boosting

[7] Verma GK, Soni B, Bourennane S, Ramos ACB. Data science: theory, algorithms, and applications. Transactions on Computer Systems and Networks. Springer (2021). doi: 10.1007/978-981-16-1681-5

[8] Rufo DD, Debelee TG, Ibenthal A, Negera WG. Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM). Diagnostics (2021) 11(19): 1714. doi: 10.3390/diagnostics11091714

[9] Liao H, Zhang X, Zhao C, Chen Y, Zeng X, Li H. LightGBM: an efficient and accurate method for predicting pregnancy diseases. Journal of Obstetrics and Gynaecology (2021) 42(4):620-629. doi: 10.1080/01443615.2021.1945006.

[10] Anitha C. Real-time social media sentiment analysis: a comparative study of LightGBM and SVM. En: 2025 Int. Conf. on Electronics and Renewable Systems (ICEARS); 2025; Tuticorin, India. p. 1947-1950. doi: 10.1109/ICEARS64219.2025.10940651

[11] Monteiro P, Lino J, Esteves Araújo R, Costa L. Comparison between LightGBM and other ML algorithms in PV fault classification. EAI Endorsed Transactions on the Energy Web (2024) 11(1):1-7. doi: 10.4108/ew.4865.

[12] Essa E, Omar K, Alqahtani A. Fake news detection based on a hybrid BERT and LightGBM models. Complex & Intelligent Systems (2023) 9:6581-6592. doi: 10.1007/s40747-023-01098-0
[13] Guo M, Guo Y, Peng Y, Zhang W, Ling Q. Fault diagnosis of bolt loosening based on LightGBM recognition of sound signal features. IEEE Sensors Journal (2023) 23(19):22777-22787. doi: 10.1109/JSEN.2023.3303223
[14] Kannapiran P, Sindha MMR. Voice-based gender recognition model using FRT and LightGBM. Tehnički vjesnik (2023) 30(1):282-291. doi: 10.17559/TV-20220302182704
[15] Xiwen Y, Xiaosong Z. Speaker recognition system with limited data based on LightGBM and fusion features. En: 6th Int. Conf. on Computational Intelligence and Applications (ICCIA); 2021; Xiamen, China. p. 160-164. doi: 10.1109/ICCIA52886.2021.00038
[16] Ong KL, Lee CP, Lim HS, Lim KM. Speech emotion recognition with light gradient boosting decision trees machine. International Journal of Electrical and Computer Engineering (IJECE) (2023) 13(4):4020-4028. doi: 10.11591/ijece.v13i4.pp4020-4028
[17] Yu J, Qu Y, Zhang Z, Lu Q, Qin Z, Liu X. Speech recognition based on concatenated acoustic feature and LightGBM model. En: Twelfth Int. Conf. on Signal Processing Systems. Proc. SPIE (2021) 11719:181-188. doi: 10.1117/12.2581426
[18] Sun X, Fu J, Wei B, Li Z, Li Y, Wang N. A self-attentional ResNet-LightGBM model for IoT-enabled voice liveness detection. IEEE Internet of Things Journal (2023) 10(9):8257-8270. doi: 10.1109/JIOT.2022.3230992
[19] Guryanov A. Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees. En: van der Aalst W, et al, editores. Analysis of Images, Social Networks and Texts. AIST 2019. LNCS. Springer, Cham (2019) 11832:39-50. doi: 10.1007/978-3-030-37334-4_4.
[20] McFee B, Raffel C, Liang D, Ellis DP, McVicar M, Battenberg E, et al. Librosa: audio and music signal analysis in Python. En: Proc. of the 14th Python in Science Conference; 2015; Austin, TX, USA. p. 18-25. doi: 10.25080/Majora-7b98e3ed-003.
[21] de Paz GGE, Quintero Flores, Quiñones Solís X. Lenguaje de programación para la composición automática de música. Programación Matemática Y Software (2025) 17(1):1-10. doi: 10.30973/progmat/2025.17.1/1
[22] Shagi GU, Aji SA. A machine learning approach for gender identification using statistical features of pitch in speeches. Applied Acoustics (2022) 185:108392. doi: 10.1016/j.apacoust.2021.108392
[23] Kone VS, Anagal A, Anegundi S, Jadhav P, Kulkarni U, MSM. Voice-based gender and age recognition system. En: Int. Conf. on Advancement in Computation & Computer Technologies (InCACCT); 2023; Gharuan, India. p. 74-80. doi: 10.1109/InCACCT57535.2023.10141801.
[24] Cortes Aguilar TA, Tovar Arriaga A. Spanish voice command recognition. Figshare. Dataset (2025). doi: 10.6084/m9.figshare.30790625
[25] Fallon J, Pylkkänen L. Language at a glance: how our brains grasp linguistic structure from parallel visual input. Science Advances (2024) 10(43):1-9. doi: 10.1126/sciadv.adr9951
[26] Malik M, Malik MK, Mehmood K, Makhdoom I. Automatic speech recognition: a survey. Multimedia Tools and Applications (2021) 80:9411-9457. doi: 10.1007/s11042-020-10073-7
[27] Batuwita R, Palade V. FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Transactions on Fuzzy Systems (2010) 18(3):558-571. doi: 10.1109/TFUZZ.2010.2042721
[28] Dumitru CO, Gavat I. A comparative study of feature extraction methods applied to continuous speech recognition in Romanian language. En: Proceedings ELMAR; 2006; Zadar, Croatia. p. 115-118. doi: 10.1109/ELMAR.2006.329528
[29] Trentin E, Gori M. A survey of hybrid ANN/HMM models for automatic speech recognition. Neurocomputing (2001) 37(1-4):91-126. doi: 10.1016/S0925-2312(00)00308-8

ACERCA DE LOS AUTORES

Teth Azrael Cortes-Aguilar (Dr.) Recibió el grado de ingeniero en comunicaciones y electrónica por la Universidad de Guadalajara en 2003. Obtuvo el grado de maestro en ciencias en óptica por el centro de investigación CICESE, Ensenada, México en 2005. En 2022 recibió el grado de doctor en manufactura avanzada por el centro de tecnología avanzada CIATEQ. Actualmente se desempeña como profesor investigador en el departamento de ingeniería electrónica, con perfil PRODEP del Tecnológico Nacional de México en la unidad académica Zapopan del Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Jalisco. Desde 2024 es candidato en el SNII de la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) del gobierno del México.

Adriana Tovar-Arriaga (MSC.) Recibió en grado de ingeniera en sistemas computacionales por el Instituto Tecnológico de Querétaro, México, en 1999, y el grado de maestría en sistemas computacionales por el Instituto Tecnológico de Zapopan en 2016. Desde 2007 es profesora de la carrera de ingeniería en sistemas computacionales con perfil PRODEP del Tecnológico Nacional de México en la unidad académica Zapopan del Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Jalisco, México. También es profesora en el departamento de sistemas computaciones en la universidad ITESO, Tlaquepaque, Jalisco, México.