



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido

Julian David Caballero López

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia
2021

Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido

Julian David Caballero López

Tesis de investigación presentada(o) como requisito parcial para optar al título de:

Magíster en Ingeniería - Automatización Industrial

Director (a):

Ph.D. Jan Bacca Rodríguez

Codirector (a):

PhD. Sergio Iván Villamizar Delgado

Línea de Investigación:

Clasificación de fonemas de habla silenciosa Grupo de Investigación:

Grupo de Investigación:

Grupo de investigación en electrónica de alta frecuencia y telecomunicaciones (CMUN)

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica

Bogotá, Colombia

2021

Después de todo, ¿qué es un científico entonces? Es un hombre curioso que mira a través del ojo de una cerradura, la cerradura de la naturaleza, tratando de saber qué es lo que sucede

Jacques Yves Cousteau

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

Julian David Caballero López

09/11/2021

Agradecimientos

Agradezco a la Universidad Nacional seccional Bogotá por facilitar un entorno adecuado para la investigación, así como a las personas que me acompañaron en este proceso: el docente Ph.D. Jan Bacca como mi director de tesis, mi codirector el Ph. D. Sergio Villamizar, y a los profesores que me brindaron sus conocimientos a través de sus enseñanzas y consejos.

A las personas de las diferentes instituciones, que estuvieron acompañándome en este proceso, con sus comentarios y recomendaciones, aportando sus conocimientos para mi crecimiento como profesional, persona e investigador.

Agradezco a mi familia por su apoyo constante, y a mi pareja, Carol Rangel, por su constante ayuda y comprensión durante este tiempo. Finalmente, agradezco a MinCiencias (Ministerios de Ciencia, Tecnología e Innovación de Colombia) – Convocatoria 777 de 2017.

Resumen

Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido

Durante el desarrollo de este proyecto se analizaron metodologías para la clasificación de fonemas del habla silenciosa basadas en EMD (Empirical mode decomposition). Este análisis tuvo lugar en la Universidad Nacional de Colombia (UNAL) sede Bogotá, con los datos de la base de datos Emotive-DB tomada con el equipo EMOTIVE EPOC +14, la cual contiene la información de 16 sujetos, mientras pensaban en los fonemas /a/, /e/, /i/, /o/, /u/, y las sílabas /fa/, /pe/, /mi/, /lo/, /ru/. En el proceso, se analizó la afectación en los resultados y el tiempo de procesamiento, en relación con las variables superposición, frecuencia de muestreo, cantidad de canales, entre otras; tras dicho análisis, se seleccionó y trató el número de canales, la distribución de electrodos y los vectores de proyección en la descomposición EMD; con lo cual se logró disminuir el tiempo de procesamiento promedio por trial de 8.73 segundos hasta 0.06 segundos, permitiendo así la posibilidad de implementarse en un sistema en línea.

Palabras clave: EMD, MEMD, SVD, PLV, LDA, CLASIFICADORES MULTICLASE.

Abstract

Algorithm for Online Classification of Silent Speech Phonemes Using an Embedded System

During the development of this project, methodologies for the classification of phonemes of silent speech based on EMD (Empirical mode decomposition) were analyzed. This analysis took place at the National University of Colombia (UNAL) in Bogotá, with data from the Emotive-DB database taken with the EMOTIVE EPOC + 14 equipment, which contains the information of 16 subjects, while they thought about the phonemes / a /, / e /, / i /, / o /, / u /, and the syllables / fa /, / pe /, / mi /, / lo /, / ru /. In the process, the effect on the results and the processing time were analyzed, in relation to the variables superposition, sampling frequency, number of channels, among others. After said analysis, the number of channels, the electrode distribution and the projection vectors in the EMD decomposition were selected accordingly. As a result, it was possible to reduce the average processing time per-trial from 8.73 seconds to 0.06 seconds, thus allowing the possibility of being implemented in an online system.

Keywords: EMD, MEMD, SVD, PLV, LDA, MULTICLASS CLASSIFIERS.

Contenido

	Pág.
Lista de figuras	XIII
Lista de tablas	XIV
Lista de Símbolos y abreviaturas	XV
Introducción	1
1. Contexto del proyecto.....	1
1.1 Adquisición y base de datos	2
1.2 Adquisición Emotive-DB	2
1.3 Adquisición Gtec-DB	3
2. Estudio de metodologías previas en CMUN.....	1
2.1 Metodología 1	1
2.1.1 Eliminación de artefactos.....	2
2.1.2 Caracterización.....	8
2.1.3 Clasificación	8
2.1.4 Resumen de resultados previos.....	11
2.2 Metodología 2	11
2.2.1 Eliminación de artefactos.....	11
2.2.2 Caracterización.....	12
2.2.3 Clasificación	12
2.2.4 Resumen de resultados previos.....	12
2.3 Análisis y selección de metodología base a optimizar	13
3. Identificación de factores de mayor incidencia en la clasificación	14
3.1 Definición de métricas	14
3.2 Captura de métricas	14
3.3 Listado de focos de investigación	17
3.3.1 Frecuencia de muestreo y submuestreo.	17
3.3.2 Vectores de proyección previa a la aplicación de EMD.....	18
3.3.3 Aplicación del proceso a base de datos de todos los sujetos en conjunto, versus, individualización por sujeto.	20
3.3.4 Cantidad de superposición de trials.	20
3.3.5 Cantidad de canales	22

3.3.6	Reductores de características.....	23
3.3.7	Clasificadores	24
3.4	Presentación de algoritmo completo	25
3.5	Métricas con algoritmo generado	26
4.	Comparación de resultados	29
4.1	Resultados del proceso	29
4.2	Resultados en equipo de computo	32
4.3	Resultados en sistema embebido	35
4.4	Análisis de resultados	37
5.	Conclusiones y recomendaciones.....	39
5.1	Conclusiones	39
5.2	Recomendaciones	40
5.3	Continuidad	40
	Bibliografía	43

Lista de figuras

Figura 1-1. Disposición de electrodos.	3
Figura 1-2. Fotografía y disposición de electrodos para base de datos Gtec-DB.	3
Figura 2-1. Esquema de la primera metodología propuesta.	1
Figura 2-2. Diagrama explicativo de flujo EMD.	3
Figura 2-3. Diagrama explicativo de flujo MEMD.	5
Figura 2-4. Distribución uniforme MEMD (<i>izq.</i>) y distribución adaptativa APIT-MEMD (<i>der.</i>)	6
Figura 2-5. Ejemplo de árbol de decisión.	8
Figura 2-6. Ejemplo de lógica de clasificación Adaboost.	9
Figura 3-1. Porcentaje de tiempo empleado por etapa	25
Figura 4-1. Gráfica de características de las clases /a/, /e/, /o/, /fa/, /ru/.	28
Figura 4-2. Matriz de confusión con las clases /a/, /e/, /o/, /fa/, /ru/.	29
Figura 4-3. Comparativa de resultados entre metodologías, por cada sujeto.	30
Figura 4-4. Comparativa de tiempos de descomposición	31
Figura 4-5. Comparativa de eficiencia media	32

Lista de tablas

	Pág.
Tabla 3-1. Métricas metodología 2.	14
Tabla 3-2. Métricas comparativas entre las frecuencias.	16
Tabla 3-3. Métricas comparativas entre métodos de descomposición.	17
Tabla 3-4. Métricas comparativas entre generalización e individualización de base de datos.	18
Tabla 3-5. Comparación de resultados basados en valores de superposición.	19
Tabla 3-6. Comparación de resultados basados en la selección de canales.	20
Tabla 3-7. Comparación de resultados con base a reductores de características	21
Tabla 3-8. Comparativa de resultados basadas en diferentes clasificadores.	22
Tabla 3-9. Métricas obtenidas con el algoritmo generado.	24
Tabla 4-1. Resultados de clasificación binaria entre 10 clases.	27
Tabla 4-2. Métricas obtenidas con el algoritmo generado en PYNQ	33

Lista de Símbolos y abreviaturas

Abreviaturas

Abreviatura	Término
<i>APIT-MEMD</i>	Adaptive projection intrinsically transformed multivariate empirical mode decomposition
<i>BCI</i>	Brain computer interface
<i>EEG</i>	Electroencephalography
<i>EMD</i>	Empirical mode decomposition
<i>EMG</i>	Electromyography
<i>IMF</i>	Intrinsic mode function
<i>MARS</i>	Multivariate adaptive regression splines
<i>MEMD</i>	Multivariate empirical mode decomposition
<i>SVD-MEMD</i>	Singular vector decomposition - multivariate empirical mode decomposition
<i>PLV</i>	Phase locked value

Introducción

La ausencia de un miembro puede derivarse de una condición genética, una enfermedad o un traumatismo. Esta condición genera dificultades al realizar tareas de la vida diaria y disminuye la calidad de vida de las personas. En Colombia la guerra ha estado presente por más de 60 años, lo cual ha derivado en un total de 11801 víctimas de minas antipersonales según cifras de 2019 [1], de las cuales se estima que un 80,5% permanecen con algún tipo de secuela física, siendo en algunos casos, amputaciones [2].

Las amputaciones a causa de traumatismos, accidentes, y principalmente, de las guerras han sido un problema atemporal. Se han encontrado soluciones protésicas desde el antiguo Egipto, donde se encontró una momia con una prótesis de dedo del pie [3]. A través de la historia el desarrollo de prótesis ha evolucionado en materiales, comodidad, pero sobre todo utilidad[4]. La guerra civil en Estados Unidos, así como la primera y segunda guerra mundial tuvieron como resultado un gran número de víctimas con amputaciones; lo anterior llevó a la fundación de la *American Prosthetics and Orthotics Association* y al gobierno de Estados Unidos a destinar fondos para el desarrollo de esta área [5]. En la actualidad el desarrollo de prótesis ha llegado a sistemas con varios grados de libertad y avanzados sistemas de control [6].

En cuanto a los sistemas de control usados para el manejo de las prótesis, se han implementado soluciones basadas en el procesamiento de señales de electromiografías (EMG). Estos sistemas captan las señales de esfuerzo producidas por músculos, las cuáles después de un proceso digital de tratamiento de señales se traducen a intenciones o clases [7]. Algunos estudios con este tipo de solución han dado resultados de clasificación sobre el 90% de asertividad [8]; esto después de un procedimiento de entrenamiento retroalimentado. Sin embargo, estos sistemas tienen problemas en el algoritmo de control a causa de la fatiga muscular producida por el continuo uso de la prótesis [9].

Similar a los sistemas de control a partir de señales EMG, están las soluciones de BCI (*Brain Computer Interface*), estos sistemas se encargan de leer señales cerebrales, para posteriormente ser interpretadas[10]. Sin embargo, las señales de este tipo presentan gran dificultad para ser clasificadas debido a su naturaleza multivariable y su baja relación señal a ruido [11,12]. Algunos estudios que han implementado BCI han logrado porcentajes de clasificación superiores al 90% [13]; el inconveniente en este caso es un extenso periodo de entrenamiento dividido en varias sesiones; sesiones durante las cuales el paciente aprende a modular la amplitud de SMR (*Sensori motor rhythms*) [14], lo que finalmente permite ese alto porcentaje en la clasificación.

Durante la investigación presentada en este documento, se estudiaron metodologías previamente implementadas en el Grupo de Investigación en Electrónica de Alta Frecuencia y Telecomunicaciones de la UNAL, también conocido como CMUN; estas metodologías realizan el procesamiento de habla silenciosa a través de sistemas BCI, teniendo como fuente señales de electroencefalograma (EEG por sus siglas en inglés – *Electroencephalography*-). Como metodologías base para el estudio se tuvieron las dos con mejores resultados hasta el momento; la metodología basada en MARS (*Multivariate Adaptive Regression Splines*) [15], y la metodología basada en MEMD (*Multivariate Empirical Mode Decomposition*) [16], más precisamente con APIT-MEMD (*Adaptive-projection intrinsically transformed multivariate empirical mode decomposition*) [17].

Para aproximar la investigación a un sistema de control de prótesis a partir de señales EEG en tiempo real, se encaminó la investigación a la adquisición de métricas como el tiempo total invertido en el procesamiento de los datos versus el porcentaje de clasificación y duración de las diferentes etapas del proceso. Durante la investigación se agregaron algunas condiciones, como la individualización de la base de datos por sujeto y la implementación de una etapa de reducción de características con el fin de mejorar el porcentaje de clasificación asertivo.

1. Contexto del proyecto

Este trabajo se desarrolló dentro del marco del proyecto “Desarrollo de una Interfaz Cerebro Computador con señales electroencefalográficas (EEG) que utilice el pensamiento del lenguaje para el control de una prótesis de miembro superior con aplicación a personas discapacitadas con amputaciones debidas al conflicto armado colombiano”, financiado por la convocatoria 777 de 2017 de MinCiencias en; el cual tiene como objetivo principal “Desarrollar una Interfaz Cerebro Computador con señales electroencefalográficas (EEG) que utilice el pensamiento del lenguaje de personas discapacitadas con amputaciones de miembro superior para el control de una prótesis de miembro superior”. Con vista en el logro de ese objetivo general, se plantaron una serie de objetivos específicos, entre los que se presentaba:

- *Contrastar los algoritmos creados en la etapa anterior para su aplicación como Interfaz Cerebro Computador para el control de una prótesis de miembro superior”.*

En cumplimiento de este último objetivo, se planteó el desarrollo de esta tesis, la cual se encargó de la optimización temporal de uno o más algoritmos desarrollados durante la primera etapa de investigación. Esto con el fin de buscar viabilidad en la implementación futura de un sistema de procesamiento de fonemas de habla silenciosa en tiempo real o en línea, acercándose así al objetivo ideal de crear un producto prótesis de miembro superior con base al procesamiento de los fonemas seleccionados a partir de señales EEG. Los títulos de los capítulos deben ser concertados entre el alumno y el director de la tesis o trabajo de investigación, teniendo en cuenta los lineamientos que cada unidad académica brinda. Así, por ejemplo, en algunas facultades se especifica que cada capítulo debe corresponder a un artículo científico, de tal manera que se pueda publicar posteriormente en una revista.

A continuación, se presenta un breve resumen de la metodología del macroproyecto presentado en la convocatoria.

1.1 Adquisición y base de datos

Durante el desarrollo investigativo del proyecto general presentado a MinCiencias, se hallaron y utilizaron múltiples bases de datos, entre ellas, la base de datos “*KARA ONE database: Phonological Categories in imagined and articulated speech*” dispuesta internacionalmente para uso investigativo por parte del *Toronto Rehabilitation Institute* [18]. Adicionalmente, durante el proceso mismo del desarrollo de la investigación, se realizó la construcción de 2 bases de datos propias del proyecto, estas 2 bases de datos se tomaron en etapas diferentes del proyecto:

- Etapa inicial del proyecto se adquirió la que será mencionada en adelante como Emotive-DB
- Etapa final, la mencionada como Gtec-DB.

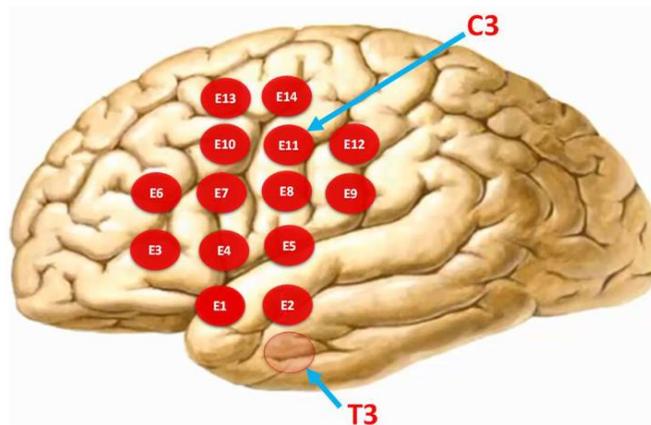
Por esta razón, el desarrollo de esta tesis se realizó con base a la Emotive-DB, sin embargo, el proceso investigativo continúa, por lo cual se presentan igualmente las características de Gtec-DB.

1.2 Adquisición Emotive-DB

La construcción de Emotive-DB consistió en la adquisición de las señales a través de 14 canales de lectura, utilizando el equipo EMOTIVE EPOC +14, el cual ofrece una frecuencia de muestreo de hasta 128Hz. Los datos consistieron en la señal adquirida a través del EMOTIVE EPOC +14 mientras el sujeto pensaba en un fonema previamente anunciado a él. Puntualmente, el proceso consistió en 20 repeticiones por cada fonema; estas repeticiones a su vez tuvieron una etapa en alto de cuatro segundos y una etapa en bajo de tres segundos. La etapa en alto fue la etapa en la cual el sujeto pensaba en pronunciar el fonema, mientras que la etapa en bajo era el periodo de descanso. Este procedimiento se realizó con las 5 vocales del español /a/, /e/, /i/, /o/, /u/ y los fonemas /fa/, /pe/, /mi/, /lo/, /ru/.

Durante el procedimiento de toma de datos, se tuvieron 16 sujetos voluntarios en plenitud de salud, todos estos diestros. A continuación, en la figura 1-1, se presenta el diagrama de la disposición de los electrodos, teniendo en cuenta que la ubicación del electrodo E2 se realizó a partir de la del electrodo T3 del sistema 10-20 con el fin de reducir la variabilidad en los datos debido a la ubicación de los electrodos entre un sujeto y otro [19].

Figura 1-1. Disposición de electrodos.



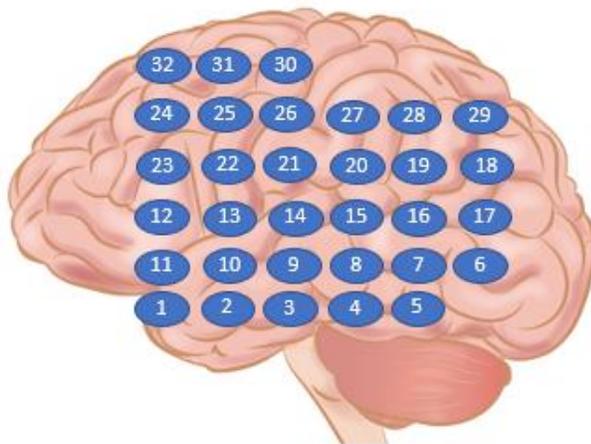
Nombre de la fuente: Tomada de Villamizar, 2021 [19].

1.3 Adquisición Gtec-DB

Esta base de datos consta de señales EEG obtenidas a partir de 32 electrodos ubicados en el hemisferio izquierdo como se muestra en la figura 1-2.

Figura 1-2. Fotografía y disposición de electrodos para base de datos Gtec-DB.

4 Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido



Nombre de la fuente: Autor.

El proceso se realizó en un entorno controlado en donde se le pedía al sujeto estar atento a un indicador LED frente a él. Se le informaba al sujeto que palabra debía pensar en la siguiente tanda de ejercicios, entre esas “abrir”, “cerrar” y “parar”. Después de ser consciente de la palabra que proseguía, se iniciaba la etapa de adquisición, donde recurrentemente un LED rojo se encendía durante 2 segundos, durante los cuales el sujeto debía mencionar la palabra; posteriormente se apagaba el LED durante 3 segundos, lo que correspondía a las etapas de descanso; finalmente, se encendía el LED con color verde durante 4 segundos, momento en el cual el sujeto debía pensar en la palabra sin pronunciarla. Entre cada uno de los instantes del LED encendido, independientemente del color, había 3 segundos de descanso el cual se indicaba con el LED apagado. El proceso se repitió 25 veces por cada palabra, para cada uno de los 32 sujetos.

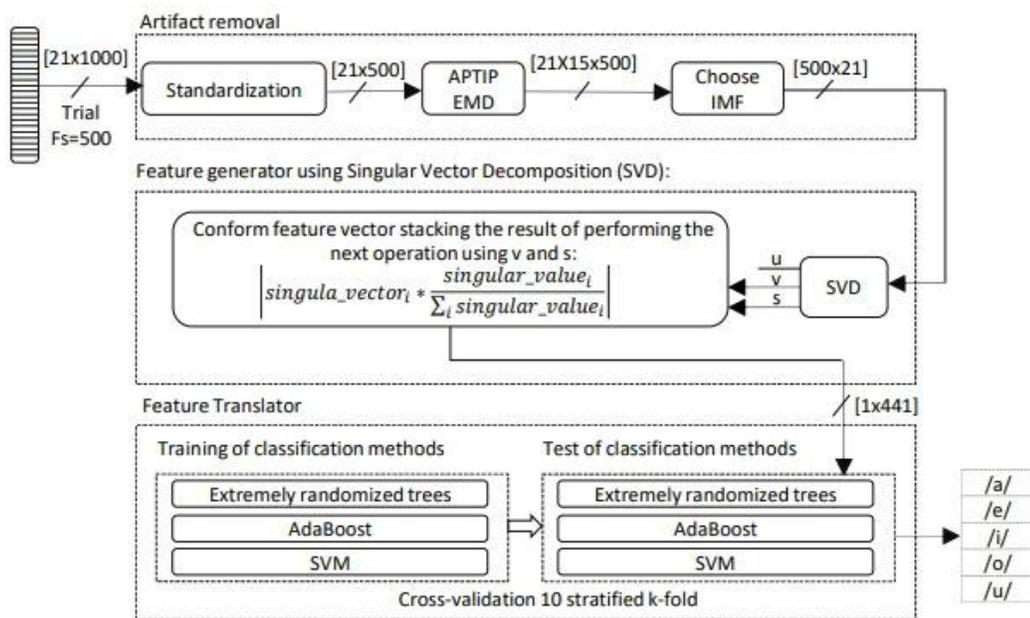
Cabe destacar que la adquisición se realizó con instrumentación de la marca *Gtec*, los equipos que se utilizaron fueron dos amplificadores *g.USBamp research* de 16 canales, electrodos *LADYbird*, dos *g.GAMMAbox*, y una *g.GAMMACAP*. Esta instrumentación se configuró para tener una frecuencia de muestreo de 2400Hz, con un filtro LPF en 2Hz, uno HPF en 200Hz y un Notch en 60Hz.

2. Estudio de metodologías previas en CMUN

2.1 Metodología 1

Esta metodología consta de una primera etapa de acondicionamiento de la señal, en la cual se realiza la eliminación de artefactos, debido a que estos no son relevantes en el procesamiento, al ser considerados como ruido. En la segunda etapa se realiza la extracción de características de la señal. Finalmente, una etapa de clasificación, la cual utiliza las características extraídas para diferenciar entre clases, que para este caso son fonemas. Se presenta entonces un diagrama representativo de la metodología, el cual se muestra a continuación en la figura 2-1:

Figura 2-1. Esquema de la primera metodología propuesta.



Nombre de la fuente: Tomada de Villamizar, 2021 [19].

2.1.1 Eliminación de artefactos

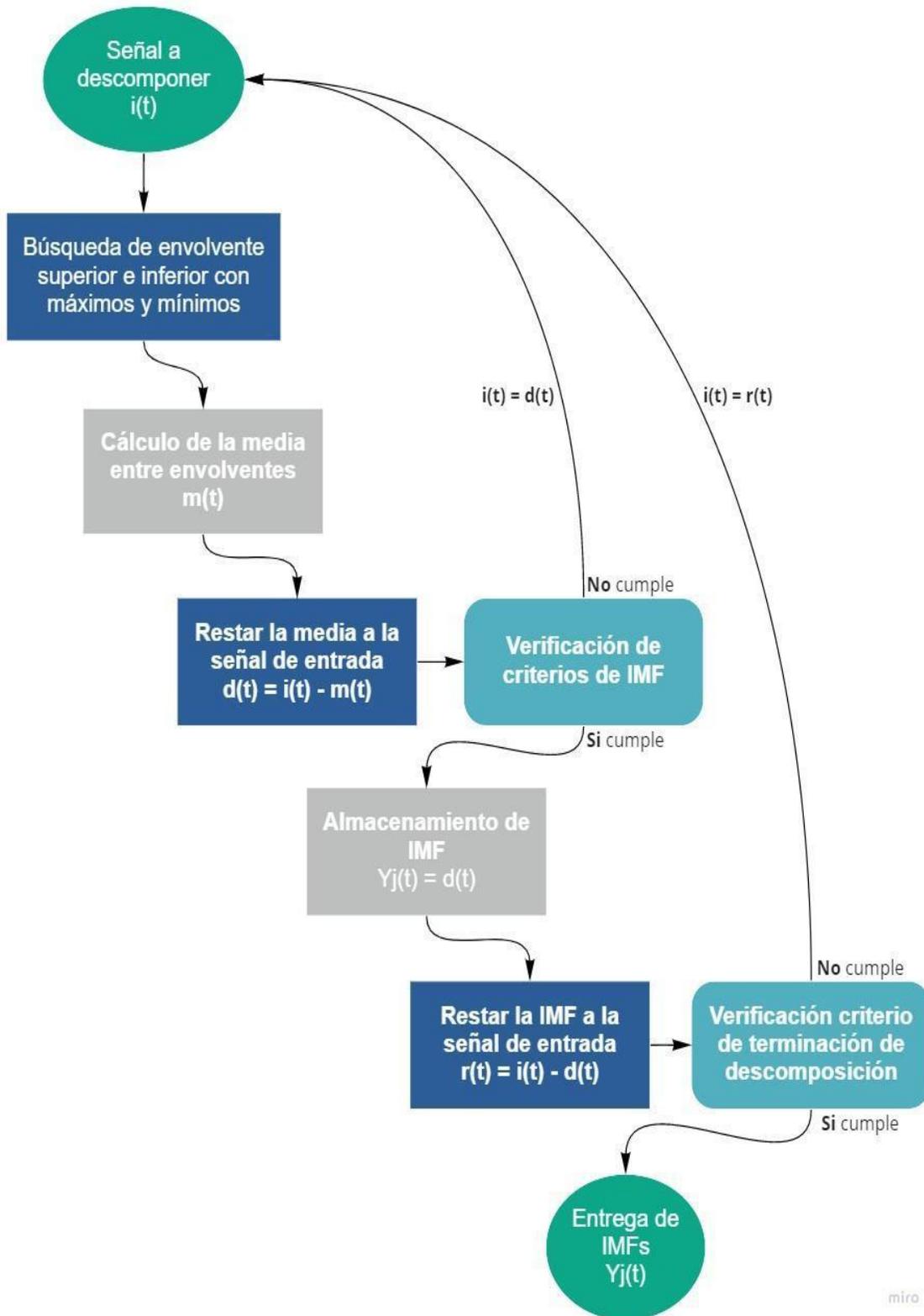
Para la eliminación de artefactos se utilizó APIT-MEMD. Este algoritmo se originó con la búsqueda de la implementación de EMD a señales multivariadas, pasando por una serie de algoritmos previos. A continuación, se aclaran los conceptos generales de EMD, MEMD y finalmente APIT-MEMD.

- Empirical mode decomposition o EMD: Este algoritmo es utilizado principalmente para la descomposición de señales no estacionarias y no lineales debido a que su descomposición es adaptativa a partir de la señal de entrada [20]. EMD surgió como base para la transformación de Hilbert-Huang, entre sus múltiples aplicaciones y características, se destaca también los resultados en el filtrado de ruido blanco [21]. El proceso de descomposición se realiza mediante la identificación de los puntos máximos y mínimos de los datos originales (señal de entrada), luego se construye una envolvente superior con los máximos y una envolvente inferior con los mínimos [22]. Posteriormente, se calcula la media entre las envolventes, la cual es restada a los datos que se tenían, y se analizan los criterios para establecer si es una función de modo intrínseco válida (IMF, por sus siglas en inglés *Intrinsic Mode function*), de no ser así, se toma este resultado como si fuese la nueva señal de entrada y se repite el proceso. En caso de ser una IMF válida, se almacena dicha información como IMF, y es restada de la señal original, para generar un nuevo punto de partida e iniciar el proceso completo nuevamente. De este modo se genera un listado de descomposición en IMFs, cada una de las cuales se encuentra en un rango de frecuencia menor a su predecesora.

Este algoritmo genera una serie de funciones derivadas llamadas IMFs, las cuales se pueden usar para la reconstrucción de la data original. Sin embargo, estas son una descomposición virtual de la señal y de ninguna manera deben considerarse como componente real de la naturaleza de la señal de entrada. A continuación, en la figura 2-2 se presenta el resumen del EMD.

Figura 2-2. Diagrama explicativo de flujo EMD.

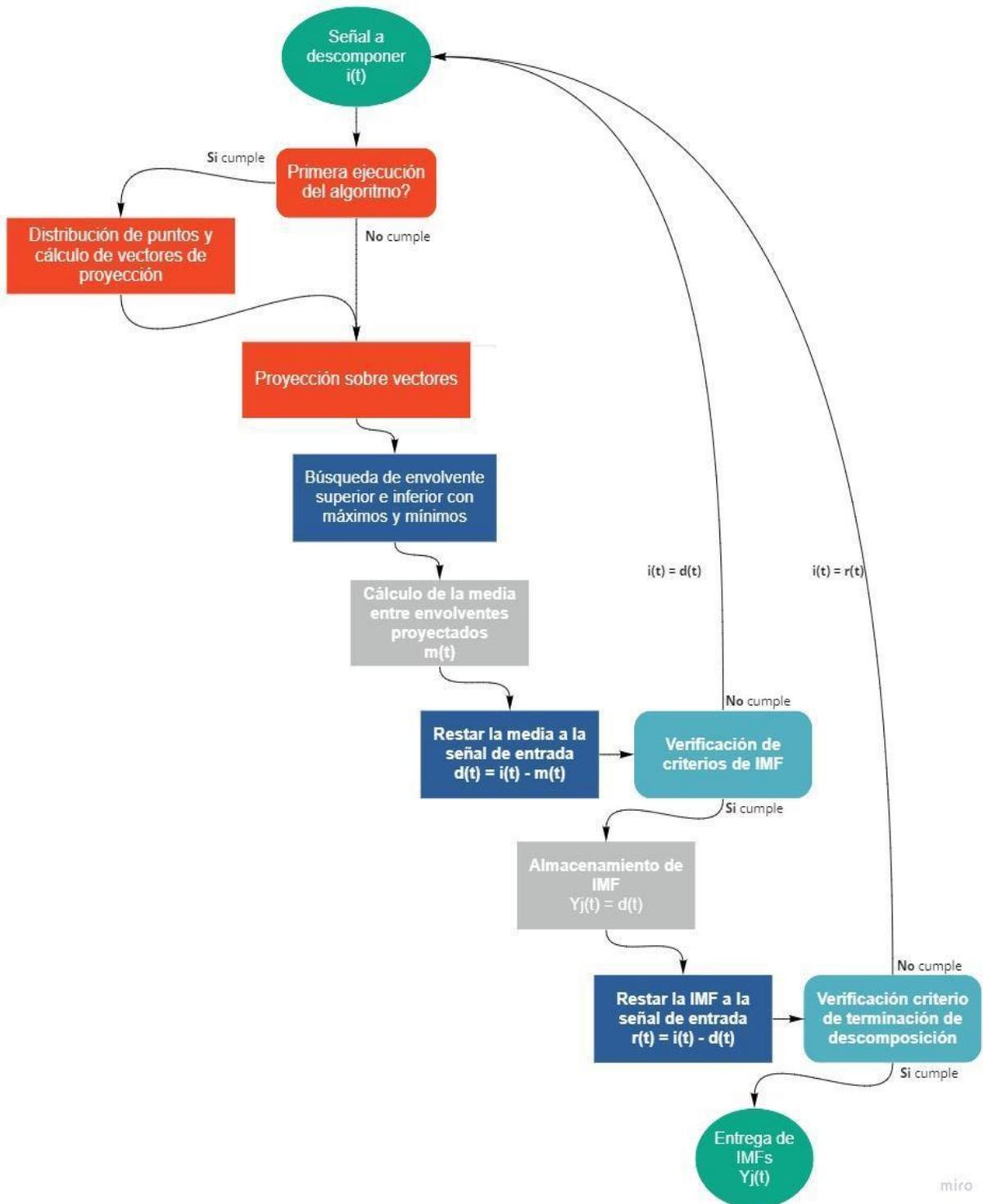
4 Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido



Nombre de la fuente: Autor.

- Multivariate empirical mode decomposition o MEMD: Este algoritmo se generó con el fin de darle aplicabilidad a la descomposición EMD con datos multivariados [16], esta necesidad se vio implícita al tratar con señales multicanal, por lo que se dieron distintas soluciones, pasando por métodos bivariados [23] y trivariados [24]. Finalmente, MEMD extiende el concepto de solución planteado en casos anteriores, el cual consiste en proyectar la señal sobre unos vectores de proyección, para calcular cada una de las envolventes, y luego promediar las mismas, obteniendo finalmente con este promedio lo que sería la media en el proceso EMD base. Después de obtener la media, el proceso continúa siendo el mismo del EMD para la extracción de cada una de las IMF. Una de las cuestiones más importantes en este método fue la selección de los vectores de proyección, los autores se inclinaron por hacer una distribución de baja discrepancia sobre una esfera n-dimensional utilizando la secuencia de Hammersley. El problema de esta solución recae principalmente en que el rendimiento decae notoriamente a medida que se aumenta el número de dimensiones. A continuación, en la figura 2-3 se presenta el resumen del MEMD.

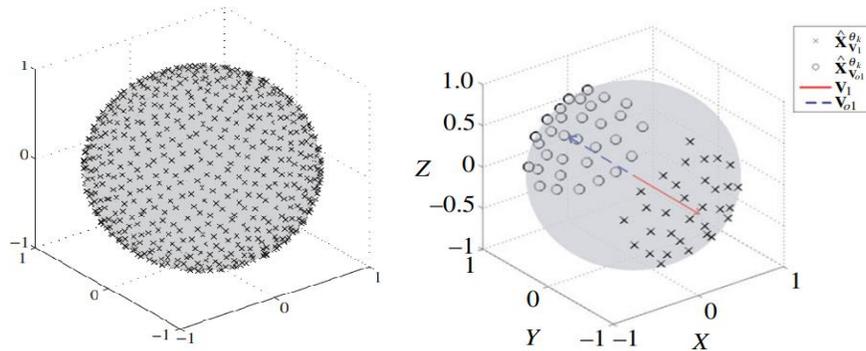
Figura 2-3. Diagrama explicativo de flujo MEMD.



Nombre de la fuente: Autor.

- *Adaptive projection intrinsically transformed multivariate empirical mode decomposition* o APIT-MEMD: Esta metodología surge a partir de MEMD, así como algunas otras variantes como X-EMD (*eXtended-EMD*) [25] creada con el enfoque de hacer filtrado multicanal, o como FEMD (*fast-MEMD*)[26] creada para disminuir el costo computacional del MEMD original al disminuir la cantidad de iteraciones. Los autores de APIT-MEMD la crean, ya que como ellos sustentan, el MEMD con su distribución uniforme, no hacen buen manejo de señales con desequilibrios de potencia y datos correlacionados entre canales, a menos que se proyecte sobre un gran número de vectores de proyección (más de 127) [17]. APIT-MEMD se trata de una modificación en la distribución de los vectores de proyección. Este método hace una descomposición en vectores y valores propios a partir de la covarianza, después de esto toman el vector principal junto con su vector diamétricamente opuesto, y finalmente trasladan los vectores de proyección restantes, acercándolos a uno de los 2 vectores principales. De esta manera se logra aumentar la eficiencia en la selección de los vectores de proyección, logrando disminuir la cantidad de proyecciones necesarias para obtener resultados acordes a lo logrado en la metodología base MEMD. A continuación, se presenta un gráfico para facilitar la comprensión de la distribución uniforme en MEMD versus la distribución reordenada con APIT-MEMD (figura 2-4).

Figura 2-4. Distribución uniforme MEMD (*izq.*) y distribución adaptativa APIT-MEMD (*der.*)



Nombre de la fuente: Tomado de Rehman & Mandic, 2019 [16] y Hemakom et al. 2016 [17].

2.1.2 Caracterización

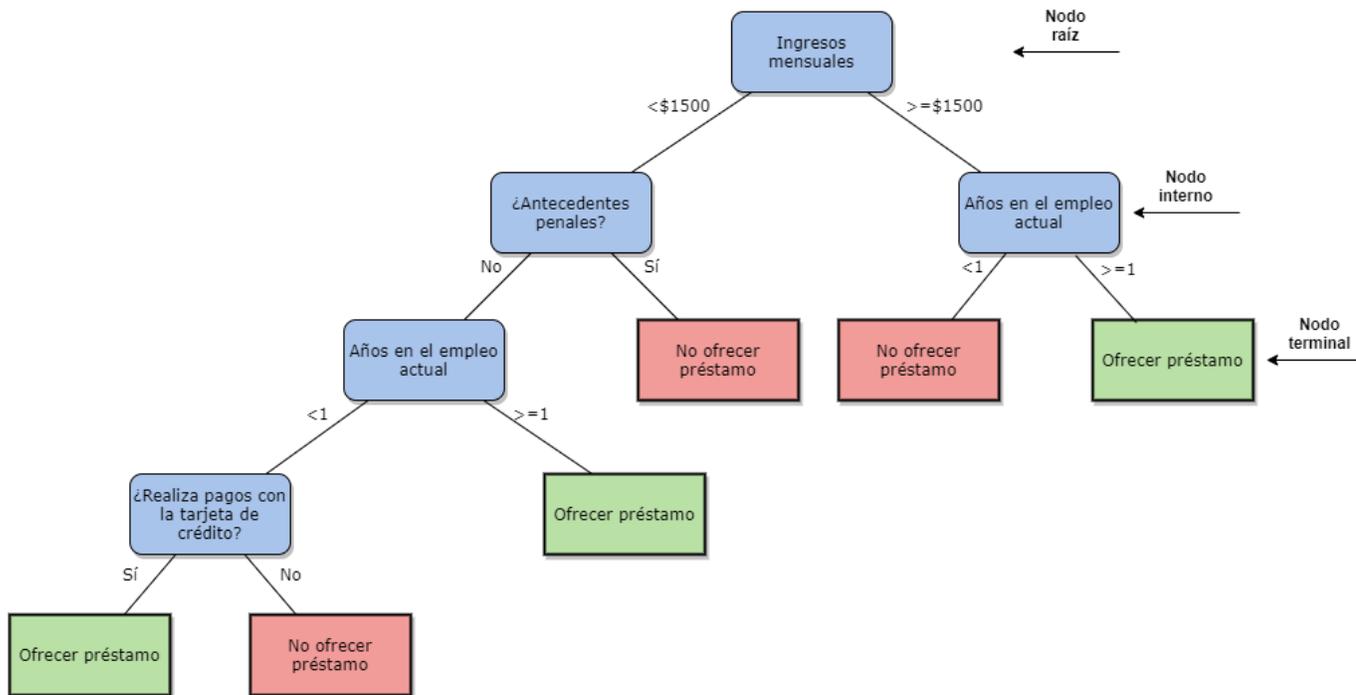
Para esta etapa se usó la herramienta de descomposición de vectores singulares o SVD (por sus siglas en inglés *Singular vector decomposition*) [27], utilizada en un sin número de aplicaciones hoy en día, desde control de procesos industriales [28], hasta la clasificación de imágenes satelitales [29]. Esto consiste en la extracción de vectores y valores singulares, los cuales tienen una relación directa con los eigenvectores y eigenvalores, los cuales han sido estudiados ampliamente junto con sus aplicaciones, por lo que se obvia su explicación en este momento. Puntualmente SVD tiene múltiples aplicaciones en el área de la ciencia de datos como lo es la reducción de dimensionalidad, la reducción de ruido en imágenes, la reconstrucción de rostros, estos ejemplos prácticos y la explicación matemática del proceso, se puede encontrar en “*Understanding Singular Value Decomposition and its Application in Data Science.*”, Reza Bagheri [30]. Para este caso la implementación de SVD se utilizó para obtener la matriz de características de cada trial la cual se tomó de la matriz de vectores singulares derechos de la descomposición.

2.1.3 Clasificación

Finalmente, en la etapa de clasificación se utilizó la biblioteca de herramientas de *Scikit-learn*. Se seleccionaron el clasificador ExtraTrees (*Extremely Randomized Trees Classifier*) y el clasificador Adaboost. El clasificador ExtraTrees, tiene como base u origen los árboles de decisión. Estos árboles se utilizan para llegar a tomar decisiones con base a la presencia o no de características o condiciones, las cuales dirigen el flujo por una rama

u otra. Esto se puede entender claramente con el siguiente ejemplo, representado en la figura 2-5:

Figura 2-5. Ejemplo de árbol de decisión.



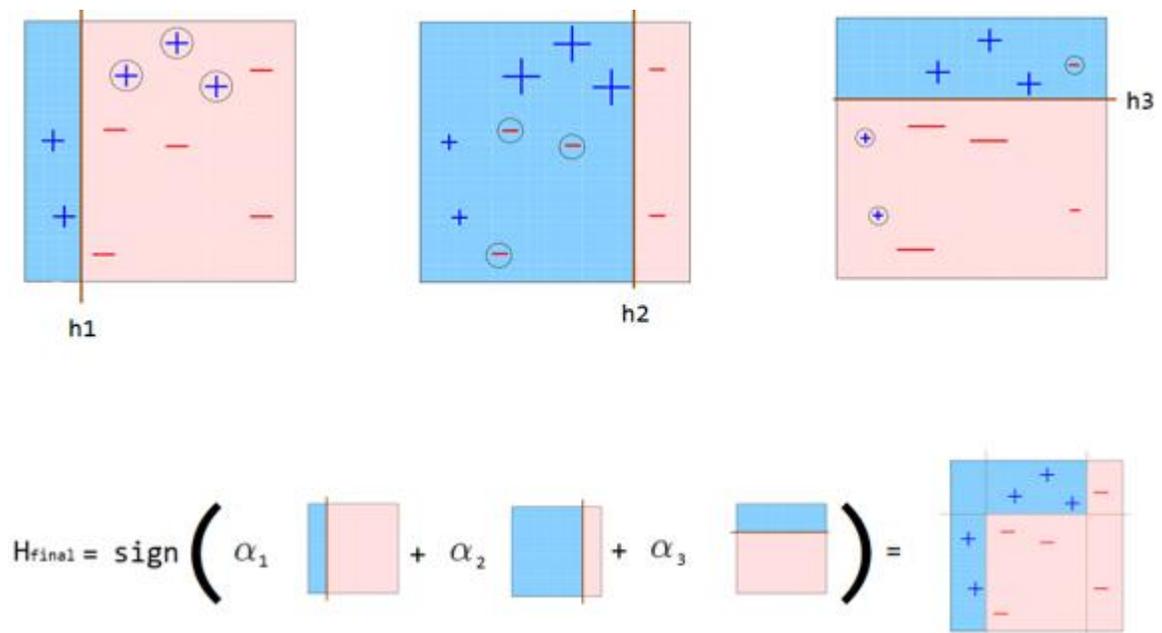
Nombre de la fuente: Tomada de DABIA, 2021. [31]

Este diagrama representa un árbol de decisión para ofrecer o no un préstamo a un cliente o empleado. Podemos observar que, según las características del cliente, el flujo de decisiones toma un camino u otro. En el caso puntual de esta investigación, las

características extraídas de la SVD, y la estructura de raíces o caminos a seguir se construyeron con base a una fracción de la base de datos de triales caracterizados. La creación de esta estructura es básicamente la etapa de entrenamiento del clasificador. Finalmente, esta estructura o clasificador entrenado, es el encargado de tomar la decisión o clasificar los triales nuevos.

Respecto al funcionamiento del clasificador Adaboost, es un clasificador que entrega una respuesta con base a una ponderación de los resultados de clasificadores menores, estos clasificadores consisten en unos pesos w_1, w_2, \dots, w_N , los cuales inicialmente se establecen en $w_i = 1/N$. Se hace una prueba de clasificación, con lo cual se revisa los casos en los que hubo error. Después se hace un ajuste de los pesos y se realiza una prueba de clasificación nuevamente, enfocando cada clasificador menor cada vez más a solventar el error de clasificación que hubo en la iteración anterior. Un ejemplo del funcionamiento se presenta a continuación en la figura 2-6.

Figura 2-6. Ejemplo de lógica de clasificación Adaboost.



Nombre de la fuente: Tomado de Freddy Hernández, 2021 [32].

De las gráficas anteriores, se puede evidenciar que el primer clasificador menor h_1 tuvo como resultado 3 errores de clasificación los cuales se muestran encerrados en círculos, en la siguiente iteración, el segundo clasificador menor h_2 se enfocó en solventar estos 3 errores que el predecesor tuvo, y resultó generando otros 3 errores de clasificación distintos; continuando con las iteraciones, el tercer clasificador menor h_3 se enfocó en clasificar bien los 3 errores cometidos en el clasificador h_2 . Finalmente, el clasificador Adaboost pondera el resultado de los pesos modelados en los 3 clasificadores menores, y finalmente da un resultado de clasificación más preciso.

2.1.4 Resumen de resultados previos

Con esta metodología se realizaron múltiples pruebas en clasificación binaria, y se obtuvo un resultado medio de hasta 92% con desviación estándar de 8% con el clasificador Extratrees. Adicionalmente, se extrajeron matrices de confusión entre los diferentes pares de fonemas, con lo que se seleccionaron 5 fonemas para pruebas de clasificación multiclase. El clasificador multiclase se construyó con los fonemas /a/ vs /e/ vs /i/ vs /o/ vs /u/, y se obtuvo un resultado de clasificación medio de 79% con desviación estándar de 7% [19].

2.2 Metodología 2

En este caso, la diferencia con respecto a la Metodología 1 radica en la etapa de extracción de características se implementa MARS y mRMR (*mínimum redundancy maximum relevance* por sus siglas en inglés) [33,34,35]; esto entrega un vector de características totalmente diferente a la etapa de clasificación y tiene un consumo de recursos más amplio.

2.2.1 Eliminación de artefactos

Para la eliminación de artefactos se inició con una estandarización de la señal de entrada, seguida de una descomposición con el uso de la metodología APIT-MEMD, y finalmente con la selección de una IMF. La selección de la IMF se realizó con base en su frecuencia (18-49 Hz), esto debido a que en este rango se han encontrado los mejores resultados en cuanto a la clasificación del habla silenciosa a partir de señales de EEG [19].

2.2.2 Caracterización

Para el modelado de señales multivariadas, una de las metodologías a utilizar es el modelo no paramétrico; debido a su característica de adaptabilidad, este resulta en una alta precisión en el modelamiento, tanto así, que en ocasiones genera problemas de sobreajuste. MARS es uno de estos métodos no paramétricos, el cual se utilizó en este caso debido a su aplicación en el modelado de datos multidimensionales. La salida del MARS fue un modelo continuo y derivable el cual posteriormente se trató con mRMR para la extracción de las características a partir de los puntos más influyentes en el comportamiento del modelo.

2.2.3 Clasificación

En la clasificación se utilizaron diferentes clasificadores con el fin de comparar sus resultados teniendo como entrada la misma matriz de características para todos los casos. Los clasificadores que fueron seleccionados fueron:

- ExtraTrees
- Adaboost
- Gaussian Naive Bayes

En esta metodología se incluyó el clasificador Gaussian Naive Bayes; este tiene algunas características importantes y deseables en la aplicación de este proyecto, entre las que se resalta que no requiere un número grande de datos de entrenamiento; adicionalmente ha demostrado obtener buenos resultados en clasificación multiclase, y finalmente con la asunción de que cada característica es independiente de la otra hace que el procesamiento sea muy ligero y sencillo de implementar.

2.2.4 Resumen de resultados previos

Se realizaron pruebas en clasificación binaria y con la base de datos KARA *One*, con los fonemas /uw/ /piy/ /diy/ /n/ /pat/ /pot/ /knew/ /gnaw/. Se obtuvieron resultados medios de 89% con desviación estándar de 7%. En cuanto a la clasificación multiclase, debido a que se habían extraído características binarias con mRMR, se realizó una concatenación de las características extraídas de las siguientes tres comparaciones binarias /pot/-vs-/gnaw/,

/pot/-vs-/knew/, y finalmente /pat/-vs-/pot. El clasificador multiclase entrenado con las clases mencionadas anteriormente entregó un resultado promedio de 86% +/- 5% con el clasificador Extratrees, 82% +/-3% con el clasificador Adaboost, y 65% +/- 4% con el clasificador Gaussian Naives Bayes [19].

2.3 Análisis y selección de metodología base a optimizar

Si se observan los resultados de clasificación de las metodologías presentadas previamente, se evidencia que hay cercanía en los mismos. Sin embargo, durante el desarrollo de las pruebas, se presentó claramente una diferencia en cuanto al tiempo que se tardaban estas metodologías. En cuanto a la metodología 1, el tiempo de procesamiento de la base de datos Emotive-DB completa estuvo alrededor de 2 horas, y por el contrario la metodología 2 tardó en el procesamiento de la misma base de datos alrededor de 6 horas. Por esta razón, se decidió tomar como punto de partida para la optimización del algoritmo la metodología 1.

3. Identificación de factores de mayor incidencia en la clasificación

3.1 Definición de métricas

Con miras al objetivo general del proyecto, y habiendo estudiado diferentes metodologías de clasificación de fonemas de habla silenciosa, se definió como principal necesidad la reducción de tiempo de procesamiento del algoritmo basado en MEMD.

Siendo la reducción de tiempo de procesamiento la principal tarea para el alcance de los objetivos, se definen entonces algunas métricas para la evaluación y estudio de los diferentes factores sobre el algoritmo de clasificación no lineal. Estas métricas se presentan a continuación:

- Tiempo de procesamiento de la base de datos completa.
- Porcentaje de clasificación media de todos los sujetos.
- Tiempo de procesamiento por sujeto.
- Porcentaje de clasificación media por sujeto.
- Tiempo de ejecución por etapa del algoritmo.
- Porcentaje de clasificación por segundo de procesamiento

3.2 Captura de métricas

Se ejecutó el algoritmo de la metodología 2, dando como resultado un promedio de clasificación acertada de 87% con una desviación estándar de 5%; esto empleando un tiempo promedio de 2838 segundos en todo su proceso. Dicho proceso se fraccionó en tres etapas con el fin de identificar el tiempo empleado en cada una de estas; en la etapa de descomposición se obtuvo un promedio de 2835 segundos, en la etapa de

caracterización el tiempo fue de 1.42 segundos y en la etapa de clasificación 2.56 segundos. Es evidente que el proceso con mayor probabilidad de ser optimizado es el de descomposición. A continuación, se presenta la tabla 3-1, en la cual se presentan los resultados obtenidos.

Tabla 3-1. Métricas metodología 2.

Sujeto	Media de clasificación [%]	Desviación estándar [%]	Tiempo en descomposición [s]	Tiempo en caracterización [s]	Tiempo en clasificación [s]	Tiempo total [s]
S1	0.86	0.06	4640.80	1.42	2.61	4644.83
S2	0.88	0.04	2897.55	1.41	2.56	2901.52
S3	0.75	0.06	2580.31	1.40	2.54	2584.25
S4	0.94	0.05	2562.61	1.37	2.52	2566.50
S5	0.89	0.04	2546.38	1.38	2.53	2550.29
S6	0.92	0.03	2610.89	1.39	2.49	2614.78
S7	0.89	0.04	2800.80	1.43	2.56	2804.78
S8	0.89	0.04	2826.53	1.42	2.52	2830.47
S9	0.91	0.06	2835.51	1.41	2.65	2839.57
S10	0.84	0.06	2629.57	1.44	2.60	2633.61
S11	0.85	0.07	2555.56	1.43	2.56	2559.56
S12	0.86	0.03	2785.15	1.44	2.56	2789.15
S13	0.89	0.07	2877.61	1.45	2.55	2881.61
S14	0.71	0.08	2938.38	1.40	2.61	2942.40
S15	0.90	0.06	2536.79	1.42	2.53	2540.73
S16	0.92	0.04	2734.93	1.46	2.54	2738.93

Promedio	0.87	0.05	2834.96	1.42	2.56	2838.94
Total			45359.38	22.67	40.93	45422.98

Adicionalmente, se resalta el tiempo empleado en la descomposición de la señal obtenida del sujeto 1; se indaga que dicho aumento notorio en el tiempo de procesamiento de la descomposición se debe a la cantidad de iteraciones requeridas por el algoritmo APIT-MEMD para identificar cada una de las IMFs. El tiempo empleado sugiere que dicho proceso de descomposición llegó a cumplir el criterio de parada por alcanzar el número máximo de iteraciones, lo cual supone que la descomposición no se llevó a cabo de manera efectiva y quedó con modas mezcladas.

Por otra parte, con el fin de tener una característica cuantitativa de eficiencia se propuso la relación entre el porcentaje de clasificación acertado y el tiempo empleado de procesamiento con el que se alcanzó dicho porcentaje. Se define entonces el término de eficiencia para este proyecto como (ver ecuación 1):

$$Eficiencia = \frac{Clasificación}{Tiempo}$$

Lo que, para efectos prácticos, y con el fin de generalizar los resultados para todos los sujetos, se expresa en función de la media y el tiempo total empleado, dando como resultado que la metodología 2 tiene como valor de Eficiencia 0.03%/s (como se presenta en la ecuación 2).

$$Eficiencia = \frac{Clasif_{media}}{Total_{tiempo}} = \frac{87\%}{2838\ s} = 0.03 \frac{\%}{s}$$

Teniendo claras las condiciones resultantes de la metodología 2, se procede entonces a investigar sobre hipótesis de variables que podrían ser ecualizadas, es decir ajustadas conjuntamente, para alcanzar valores de eficiencia que demuestren mejores resultados.

3.3 Listado de focos de investigación

A partir de la información adquirida respecto a las metodologías 1 y 2, y más precisamente los resultados de tiempo presentados implementando la metodología 2, se plantea el siguiente listado de focos de investigación:

1. Vectores de proyección previa a la aplicación de EMD.
2. Aplicación del proceso a base de datos de todos los sujetos en conjunto, versus, individualización por sujeto.
3. Cantidad de superposición de triales.
4. Cantidad de canales
5. Reductores de características
6. Clasificadores

Se realizaron una gran cantidad de experimentos tomando tiempos de ejecución y resultados de clasificación con diferentes valores de las diferentes variables por separado, y algunos con ajustes en paralelo de las variables. Estos resultados se presentan de manera resumida en el siguiente numeral:

3.3.1 Frecuencia de muestreo y submuestreo.

El primer factor analizado fue la influencia de la frecuencia de muestreo y la implementación de submuestreo de la señal en el resultado de clasificación y el tiempo empleado en el procesamiento de los datos.

Debido a que la base de datos EMOTIVE-DB se capturó a una frecuencia de muestreo de 128Hz, se tomó este valor como tope y se realizó un submuestreo a 64Hz y otro a 32Hz a modo estudio experimental, previendo que a 32Hz estaremos probablemente perdiendo información de interés. Con cada uno de los casos, se tomaron los resultados de clasificación, el tiempo de procesamiento y finalmente el cálculo de la eficiencia con base a los anteriores valores. A continuación, se presenta la tabla sintetizada de los resultados.

Tabla 3-2. Métricas comparativas entre las frecuencias.

Frecuencia [hz]	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
32	0.52	0.05	1629.72	0.032
64	0.80	0.06	2356.67	0.034
128	0.87	0.05	2838.94	0.030

Con los resultados presentados de la tabla anterior, se puede observar que la mayor eficiencia fue obtenida con una frecuencia de muestreo de 64Hz, seguida de la obtenida con 32Hz y por último 128Hz. Sin embargo, observando el promedio de clasificación, observamos que los resultados decaen 7% para el de 64Hz y 35% para el ensayo de 32Hz. Siendo 52% una media de clasificación muy baja respecto a los resultados originales se decidió descartar ya que evidencia la pérdida notoria de información de interés para la clasificación de fonemas del habla silenciosa. En cuanto al caso de 64Hz, se analiza que el tiempo disminuyó aproximadamente 500 segundos, lo que equivale a una reducción del tiempo cercana al 17% a costa de 7% en la clasificación media. Por esta razón se define como viable la aplicación de un submuestreo a 64Hz, pero se decide continuar con los experimentos con la frecuencia original de muestreo a 128Hz al tener en cuenta que el proceso de submuestreo igual abarca una tarea adicional en el proceso, volviendo a aumentar parte del tiempo de procesamiento sin recuperar porcentaje de clasificación.

3.3.2 Vectores de proyección previa a la aplicación de EMD.

Se realizó un estudio de la incidencia del número de vectores de proyección sobre el tiempo de ejecución y el porcentaje de clasificación, reafirmando lo sucedido con MEMD, donde el aumento del número de vectores produce una gran incidencia en el rendimiento del algoritmo de descomposición. Se planteó entonces una reducción notoria en la cantidad de vectores de proyección, lo cual llevó a la idea de utilizar SVD (*singular vector decomposition*). Con la descomposición SVD, se obtuvo la matriz de vectores singulares y sabiendo que la mayor parte de la información se encuentra sobre los primeros vectores

singulares de la matriz, se decide seleccionar los tres principales vectores singulares como vectores de proyección. Con esta decisión se evidencia que la descomposición MEMD pierde precisión, ya que se genera una mayor mezcla de modas en las IMFs extraídas. Sin embargo, con base a la investigación previa, durante el desarrollo del proyecto presentado a MinCiencias [19], donde se establece que el rango de frecuencias de interés va desde 16Hz a 45Hz, se establece como viable el resultado de la descomposición; lo anterior debido a que la segunda o tercera IMF obtenida siempre tiene su información en este rango de frecuencia. Se presenta entonces la tabla 3-3, donde se evidencia una comparativa de los resultados contrastados con respecto a la metodología original APIT-MEMD.

Tabla 3-3. Métricas comparativas entre métodos de descomposición.

Método	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
APIT-MEMD	0.87	0.05	2838.94	0.030
SVD-MEMD	0.72	0.05	132	0.545

De la tabla anterior se observa una disminución de 15% en el porcentaje de clasificación, lo cual es un resultado considerablemente negativo; por otra parte, vemos una reducción del tiempo de procesamiento muy notoria, lo que de alguna manera compensa la pérdida del porcentaje de clasificación. Esto se evidencia al observar el valor de la variable eficiencia que se escogió con anterioridad, donde el resultado SVD-MEMD es muy superior de entrada respecto al APIT-MEMD. Por lo anterior, la investigación se desarrolló sobre esta nueva metodología de descomposición, buscando mejorar el valor de la eficiencia, recuperando el porcentaje de clasificación acertada sacrificado y de ser posible, reduciendo aún más los tiempos de ejecución del algoritmo completo.

3.3.3 Aplicación del proceso a base de datos de todos los sujetos en conjunto, versus, individualización por sujeto.

Este foco de investigación se presentó a causa de lo evidenciado en otras publicaciones como [12,14,18], donde se expone una mejoría en los resultados de clasificación a causa de un periodo de entrenamiento propio del sujeto individualmente. Esto sugirió que se podía mejorar el porcentaje de clasificación general, individualizando la base de datos por sujeto y ejecutando el algoritmo completo, es decir: descomposición, extracción de características, entrenamiento del clasificador y finalmente prueba de clasificación. En la tabla 3-4 puede evidenciar las comparativas entre los datos analizados.

Tabla 3-4. Métricas comparativas entre generalización e individualización de base de datos.

Método	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
DB-completa	0.72	0.05	132	0.545
DB-individualizada	0.77	0.7	129	0.59

En la tabla anterior se pudo apreciar, una recuperación del 5% del porcentaje de clasificación medio, logrando nuevamente mejorar el valor de la eficiencia. Sin embargo, no se puede perder de vista que la desviación estándar aumentó 2%, lo cual representa mayor inestabilidad en los resultados de la clasificación.

3.3.4 Cantidad de superposición de triales.

Durante el proceso investigativo y de estado del arte, se encontraron algunos estudios donde se aplicaba un porcentaje de superposición al fraccionamiento de los triales de tiempo y publicaciones en las cuales se estudia este fenómeno [36,37,38]. Se estudió esta posibilidad debido a la incertidumbre existente en cuanto al momento exacto en el que los

sujetos comenzaban o interrumpían la tarea de pensar en cada fonema además de la longitud de dicho pensamiento. Al dividir la información en triales equidistantes y de igual longitud (1s) se caía en la suposición de que cada uno contenía la información distribuida de igual manera, lo cual no se podía afirmar de ninguna manera. Se seleccionó entonces 50% y 75% como porcentajes de superposición a probar. La recopilación de información bajo estas condiciones se muestra a continuación, en la Tabla 3-5.

Tabla 3-5. Comparación de resultados basados en valores de superposición.

Superposición	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
0%	0.77	0.12	129	0.59
50%	0.82	0.09	215	0.381
75%	0.87	0.05	399	0.218

Observando la tabla presentada, se encuentra el porcentaje de clasificación aumentaba proporcionalmente con respecto al porcentaje de superposición, llegando incluso a alcanzar un resultado de 87% de clasificación, lo cual supondría la recuperación del porcentaje de clasificación sacrificado al disminuir la calidad de la descomposición de IMF con la metodología SVD-MEMD. A pesar de esto, se decidió continuar el proceso de investigación con 50% de superposición, ya que con este porcentaje se aseguró que cada trial comenzara exactamente 500ms después del inicio del anterior, periodo de tiempo suficiente para iniciar y culminar la tarea de pensar en un fonema. Se descarta entonces la superposición del 75% por aumentar la posibilidad de la duplicación de información, sin embargo, no se consideró errónea la idea de tomar 75% de superposición ya que en 250ms aún es posible llevar a cabo la tarea de iniciar y terminar de pensar en un fonema, pero con un espacio para desfase demasiado reducido.

3.3.5 Cantidad de canales

En el proceso de descomposición MEMD, uno de los parámetros claves, es la relación de la cantidad de dimensiones/canales con la cantidad de vectores de proyección. Tanto los autores de APIT-MEMD [17], como los autores en MEMD [16] consideran que se debe tener un número de proyecciones mayor al número de dimensiones; sin embargo, al decidir utilizar tan sólo tres vectores de proyección en SVD-MEMD, se tiene una relación 3/14 con respecto a los 14 canales de entrada. Se plantea entonces la reducción de la cantidad de canales de entrada, aumentando la relación actual de 3/14, esperando así por una parte reducir el tiempo de procesamiento al requerir menos proyecciones, y aumentar el porcentaje de clasificación debido a la mejoría de la relación vectores de proyección por canal de entrada.

En la siguiente tabla se presentan los resultados obtenidos después de varios ensayos reduciendo y agrupando los canales de forma iterativa. Se probaron todas las combinaciones posibles, haciendo grupos desde 4 canales, hasta grupos de 10 canales. Se muestra entonces los valores promedios obtenidos de la agrupación con mejores resultados en la Tabla 3-6, siendo comparado con los resultados obtenidos con todos los canales activos.

Tabla 3-6. Comparación de resultados basados en la selección de canales.

Grupo de canales	Clasificación media [%]	Desviación estándar [%]
All (14)	0.82	0.09
[1,3,9,11,13]	0.73	0.08

A pesar de lograr una eficiencia mayor con la combinación de canales (1,3,9,11,13), es notorio que el porcentaje de clasificación se redujo nuevamente 9% en promedio. Esto supondría volver a los resultados iniciales, habiendo implementado ya los principales ajustes planteados, lo cual dificulta la tarea de recuperar puntos en el porcentaje de clasificación. Por lo anterior, se descartó la idea de la reducción de canales en esta ocasión.

3.3.6 Reductores de características

En procesos como la clasificación de imágenes, se suele utilizar algoritmos para la reducción de características. Estos se encargan de extraer un nuevo grupo de características de menor cuantía, pero este grupo, presenta una diferenciación mayor entre los valores de las clases. Para la experimentación se utilizó LDA (*linear discriminant analysis*) y PCA (*principal component analysis*), los cuales habían sido utilizados en trabajos previos, dando resultados comparativos y aumentando el porcentaje de clasificación. Esta comparación de resultados se presenta en la Tabla 3-7.

Tabla 3-7. Comparación de resultados con base a reductores de características

Reductor	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
None	0.82	0.09	215	0.381
LDA	0.89	0.06	216	0.412
PCA	0.81	0.08	215	0.376

Observando la tabla comparativa entre los resultados aplicando LDA o PCA como reductores de características versus los resultados previos, podemos concluir en una selección evidente. Con la implementación de LDA como reductor, obtenemos un resultado de clasificación 7% más efectiva, al mismo tiempo que el tiempo de procesamiento sólo se ve afectado en 1 segundo, lo cual es poco significativo. Se escoge entonces implementar LDA para la reducción de características para el algoritmo construido.

3.3.7 Clasificadores

Finalmente, la última etapa del procesamiento, la cual consiste en el entrenamiento de un modelo clasificador con una fracción de los datos, para luego ser puesto a prueba con la fracción restante. En esta etapa se utilizan las herramientas dispuestas de manera libre por parte de Scikit-Learn. La librería sklearn tiene múltiples algoritmos, entre estos, una amplia gama de clasificadores. A pesar de existir diferentes guías para la selección y configuración de clasificadores, se realizaron múltiples pruebas con las configuraciones generales de un listado de hasta 10 clasificadores. En la tabla 3-8 se resumen los resultados obtenidos.

Tabla 3-8.Comparativa de resultados basadas en diferentes clasificadores.

Clasificador	Clasificación media [%]	Desviación estándar [%]	Tiempo total [s]	Eficiencia [%/s]
Extratrees	0.89	0.06	216	0.41
Kneighbors	0.9	0.05	213	0.42
Adaboost	0.81	0.08	212	0.38
Gaussian	0.91	0.05	214	0.43
Bagging	0.89	0.06	209	0.43
Stacking	0.9	0.06	214	0.42
Gradient boosting	0.9	0.06	216	0.42
OnevsOne	0.91	0.04	211	0.43
Outputcode	0.86	0.11	211	0.41
Mlp	0.91	0.06	217	0.42

Con la tabla anterior podemos decir que, de forma muy general, el resultado de clasificación obtenido de los clasificadores tiene una varianza bastante baja. A pesar de esto, se resaltan con los mejores resultados el clasificador Gaussian, el Bagging y finalmente el OnevsOne, siendo estos tres los que lograron un valor de eficiencia de 0.43.

3.4 Presentación de algoritmo completo

Finalmente se hace una recopilación de los ajustes seleccionados durante la investigación, y se programa un algoritmo teniendo en cuenta los valores obtenidos experimentalmente. A continuación, se listan dichos ajustes:

- Frecuencia de muestreo / submuestreo: 128 Hz
- Porcentaje de superposición: 50%
- Cantidad de canales: 14
- Método de procesamiento: SVD MEMD
- Método de caracterización: PLV
- Selección de características: Reductor LDA.
- Modelo de clasificador: Multiclase, OneVsOne con LinearSVC como estimador.

Este algoritmo se implementó bajo el lenguaje de programación Python, usando como interfaz de programación y ejecución *Jupyter-Notebook*. Se creó un *notebook* para cada uno de los procesos internos del algoritmo, con los cuales se realizaron pruebas individuales y se contrastaron con resultados previos y análisis matemáticos, con el fin de asegurar el correcto funcionamiento de cada sección de forma individual. Posteriormente, se agrupó el algoritmo completo en un *notebook*, y se corrieron de nuevo las pruebas con parte de la información de la base de datos EMOTIVE-DB. Se compararon resultados, y se revisó el código en busca de posibles sesgos. Finalmente, se generaron diferentes clases en Python para optimizar la ejecución y el mantenimiento del algoritmo, y se llevó a la creación de un módulo de clasificación robusto, el cual importa las clases previamente generadas.

3.5 Métricas con algoritmo generado

Después de superar las diferentes pruebas individuales y conjuntas, se procedió a la ejecución de nuevas pruebas de clasificación para la obtención de los resultados evaluativos a través de los cuales se caracterizó los algoritmos previamente creados. El resumen de los resultados se presenta en la tabla 3-9:

Sujeto	Media de clasificación [%]	Desviación estándar [%]	Tiempo en descomposición [s]	Tiempo en caracterizar [s]	Tiempo en clasificar [s]	Tiempo total [s]
S1	0,83	0,07	10,72	0,86	0,42	11,99
S2	0,96	0,04	12,06	0,91	0,43	13,40
S3	0,80	0,09	11,47	0,88	0,42	12,77
S4	0,98	0,04	11,00	0,92	0,40	12,33
S5	0,95	0,04	10,68	0,88	0,45	12,01
S6	0,94	0,07	11,48	0,88	0,44	12,81
S7	0,98	0,03	12,12	0,89	0,44	13,44
S8	0,99	0,03	12,98	0,93	0,44	14,35
S9	0,94	0,07	13,15	0,92	0,46	14,54
S10	0,81	0,11	12,81	0,86	0,40	14,08
S11	0,91	0,05	12,06	0,91	0,44	13,41
S12	0,91	0,06	12,85	0,92	0,49	14,26
S13	0,93	0,07	13,78	0,93	0,44	15,14
S14	0,90	0,06	15,54	0,88	0,40	16,82
S15	0,93	0,08	11,63	0,87	0,45	12,95
S16	0,92	0,06	11,82	0,87	0,40	13,09

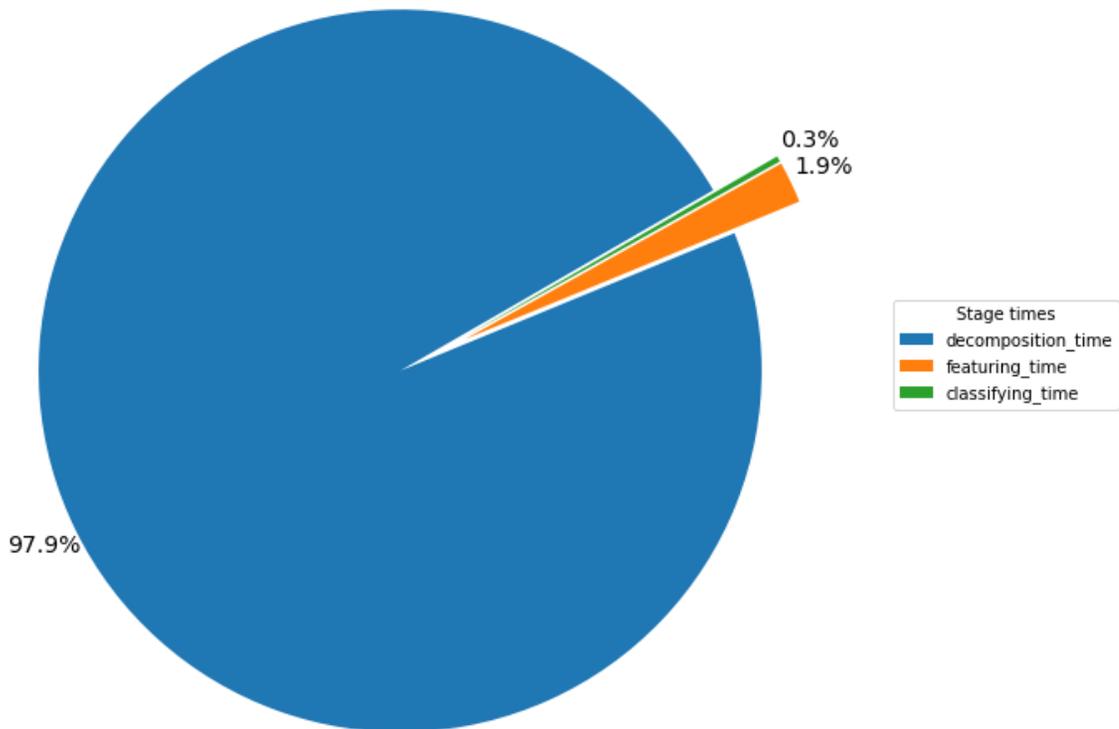
Promedio	0,92	0,06	12,26	0,89	0,43	13,59
Total			208,40	15,21	7,37	230,98

Tabla 3-9. Métricas obtenidas con el algoritmo generado.

Observando la tabla de resultados con el algoritmo generado a través de la investigación realizada durante esta tesis, se encontró que, en promedio, se obtuvo un resultado de clasificación asertiva de 92% con una desviación estándar de +/- 6%; logrando este resultado con un tiempo de procesamiento medio por sujeto de 13,59 segundos. Esto nos entrega como valor de eficiencia 6.77 %/s en promedio.

Por otra parte, analizando los resultados del tiempo empleado en cada una de las etapas principales del proceso, sigue siendo muy notorio que el algoritmo sigue tardando principalmente en el proceso de descomposición. Esto se hace evidente con la presentación de la figura 3-1 donde se evidencian los porcentajes de tiempo de cada etapa:

Figura 3-1. Porcentaje de tiempo empleado por etapa



Nombre de la Fuente: Autor.

Con los anteriores resultados, se puede entender que hubo una optimización en el proceso de clasificación de los datos presentes en la base de datos EMOTIVE-DB, pero con el fin de dar mayor claridad a los resultados logrados con el extenso proceso investigativo, se presenta en el siguiente numeral una serie de comparativas entre el punto de partida, y el logro alcanzado.

4. Comparación de resultados

4.1 Resultados del proceso

Durante el proceso investigativo del proyecto se tuvieron algunos resultados previos los cuales condujeron a la tomar decisiones sobre las cuales se continuaron los ensayos. Uno de estos, fue la selección de los fonemas a evaluar en una instancia media, e instancia final del proyecto. En la tabla 4-1 se presenta un promedio de resultados de clasificación binaria entre las 10 clases disponibles, /a/, /e/, /i/, /o/, /u/, /fa/, /pe/, /mi/, /lo/, y /ru/.

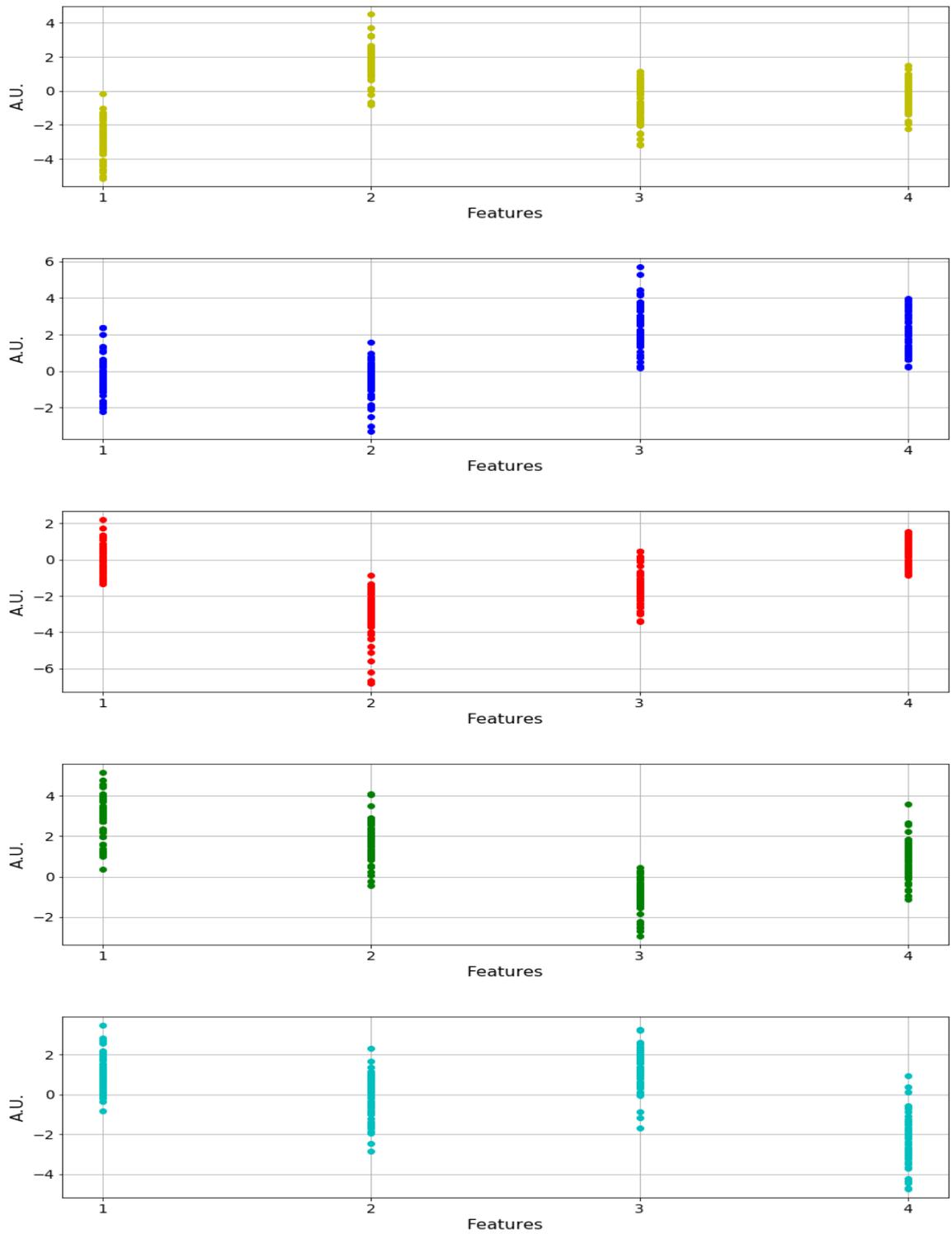
Tabla 4-1. Resultados de clasificación binaria entre 10 clases.

Clases	/a/	/e/	/i/	/o/	/u/	/fa/	/pe/	/mi/	/lo/	/ru/
/a/		0,92	0,94	0,94	0,94	0,95	0,94	0,95	0,95	0,96
/e/			0,93	0,93	0,94	0,95	0,94	0,94	0,95	0,95
/i/				0,51	0,90	0,91	0,92	0,92	0,94	0,93
/o/					0,89	0,91	0,92	0,92	0,94	0,93
/u/						0,90	0,91	0,91	0,93	0,93
/fa/							0,88	0,90	0,91	0,92
/pe/								0,88	0,90	0,89
/mi/									0,89	0,90
/lo/										0,86
/ru/										

Estos resultados de clasificación se obtuvieron con un porcentaje de superposición de triales del 75%, con el método de descomposición a través del criterio SVD-MEMD, la caracterización a partir de PLV, implementando un clasificador Extratrees y promediando finalmente los resultados obtenidos para todos los sujetos.

Observando el resultado presentado en la Tabla 4-1, se realizó pruebas con un clasificador multiclase de 5 clases, tomando /a/, /e/, /o/, /fa/, /ru/ como las clases de prueba. Con estas clases se ahondó en la investigación, y se sacó el gráfico de características presentado en la figura 4-1, así como la matriz de confusión que se presentará en la figura 4-2.

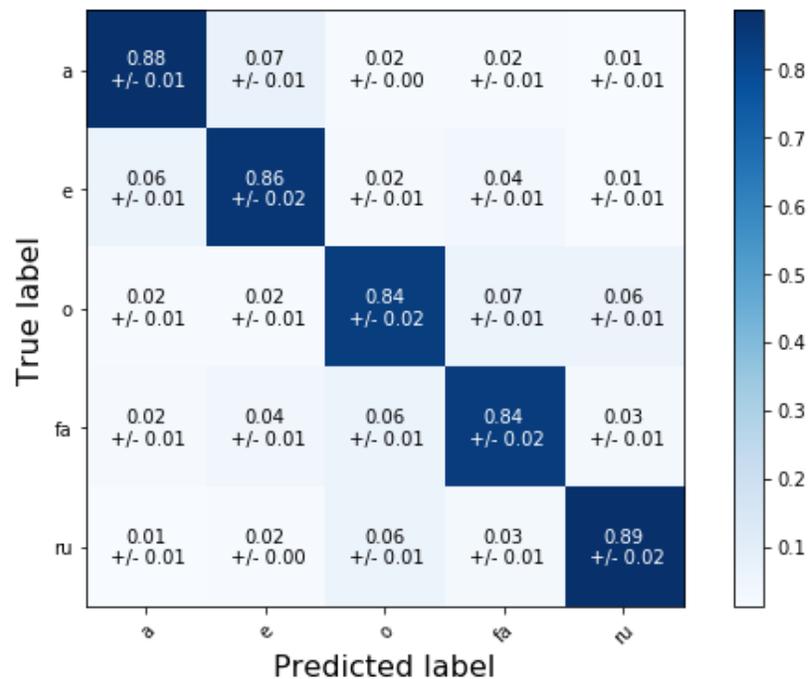
Figura 4-1. Gráfica de características de las clases /a/, /e/, /o/, /fa/, /ru/.



Nombre de la Fuente: Autor.

Esta figura, presenta las 4 características extraídas usando PLV y el reductor LDA. Donde cada gráfico interno representa cada una de las clases /a/, /e/, /o/, /fa/, /ru/ respectivamente. Los puntos demarcados en las gráficas representan los valores de cada trial correspondiente a esa clase. Es decir, para la clase /a/ observamos que la característica 1 tiene valores que oscilan entre -1 y -5, a diferencia de la clase /fa/, en la cual los valores de esta misma característica oscilan entre 1 y 5. Lo anterior significa que, si tuviésemos un clasificador binario entre la clase /a/ y la clase /fa/, teniendo en cuenta únicamente la característica 1, tendríamos unos resultados de clasificación correcta bastante altos. Sin embargo, si se usara esta misma característica para la distinción entre las clases /o/ y /ru/, se obtendría un resultado muy cercano a la aleatoriedad.

Figura 4-2. Matriz de confusión con las clases /a/, /e/, /o/, /fa/, /ru/.



Nombre de la Fuente: Autor.

Con las condiciones de procesamiento mencionadas al inicio de este numeral, se extrajo la matriz de confusión presentada en la figura 4-2. Esta matriz de confusión presenta una guía muy útil para determinar si los resultados promedios de clasificación obtenida, presenta valores desbalanceados entre la distinción de las clases. Un ejemplo de lo que sería un resultado desbalanceado, sería en el cual la matriz de confusión presentara un

valor de la diagonal con una diferencia notoria de los demás, es decir, por ejemplificar en este caso, si el 4 valor de la diagonal no fuese 0.84 sino 0.74. Lo anterior, daría a entender que la clase /fa/ en realidad no sería una buena opción para el clasificador multiclase, con lo cual debería ser reemplazada. En el caso puntual de la figura 4-2, lo que interpretamos es una correcta selección de las clases a utilizar en el clasificador.

4.2 Resultados en equipo de computo

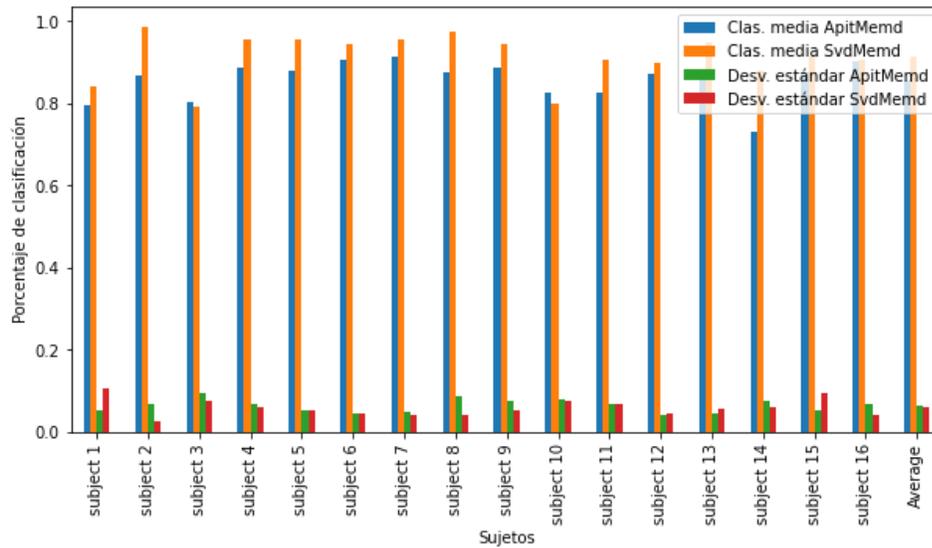
En este numeral se presentarán unos gráficos comparativos entre los resultados previos versus los resultados alcanzados con el proceso investigativo. Primeramente, se muestra la información obtenida utilizando el equipo de cómputo dispuesto en el laboratorio de la Universidad Nacional de Colombia. Este equipo consta de las siguientes características principales:

- Procesador Intel I7.
- Memoria RAM con 32GB.
- Sistema operativo Windows 10.
- Almacenamiento 1TB SSD.
- GPU Nvidia Titan X.

Las condiciones bajo las cuales se utilizó este hardware fueron por defecto, sin embargo, se aclara que se podrían lograr mejores resultados en cuanto al rendimiento con el hardware disponible, si se hace una configuración más exhaustiva para el aprovechamiento de la capacidad de este. Esto se omitió del proyecto, debido a que se esperaba implementar el algoritmo en un equipo móvil, el cual, de antemano, presentaría unas capacidades de hardware bastante más reducidas.

Se presenta entonces la figura 4-3 mostrando una comparación entre los valores medios de clasificación y desviación estándar por sujeto con el algoritmo ApitMemd en contraste con el algoritmo SvdMemd. En dicha figura, se puede observar una tendencia a mantener los resultados de clasificación similares, pero como aspecto negativo, la desviación estándar del algoritmo con SvdMemd es mayor en la mayoría de los casos.

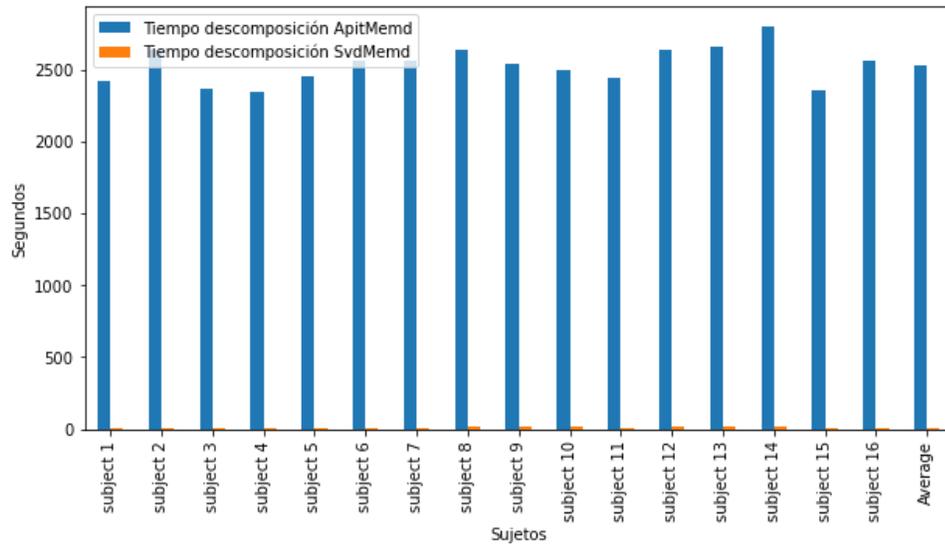
Figura 4-3. Comparativa de resultados entre metodologías, por cada sujeto.



Nombre de la Fuente: Autor.

Posteriormente, se presenta una gráfica comparativa de los tiempos de ejecución de los 2 algoritmos en cuestión. Podemos destacar que la diferencia es tan grande, que, al poner los resultados en una misma escala, el tiempo de la descomposición SvdMemd se vuelve difícil de observar, así como su variación entre cada uno de los sujetos. Esto se evidencia en la figura 4-4 a continuación.

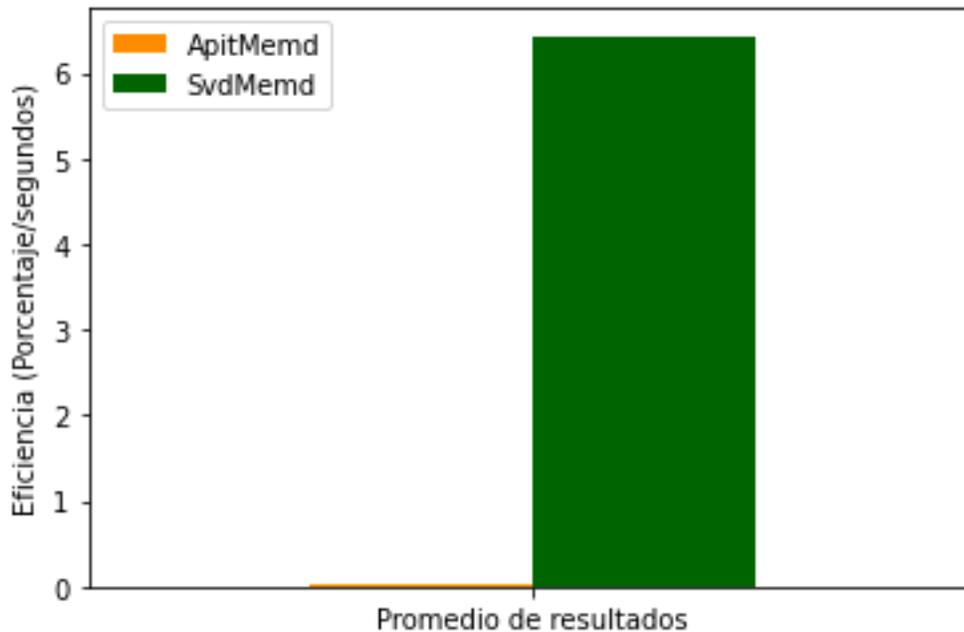
Figura 4-4. Comparativa de tiempos de descomposición



Nombre de la Fuente: Autor.

Por último, se presenta la comparativa entre los valores de la eficiencia medida para este proyecto, recordando que es un valor representativo fraccionando el porcentaje de clasificación acertado sobre la cantidad de segundos de procesamiento empleado para obtener dicho porcentaje. La figura 4-5 representa, de cierta manera, una optimización en la clasificación del habla silenciosa; esto, recordando que se hizo para unas clases seleccionadas, las cuales son dadas por los fonemas del español /a/, /o/, /ru/.

Figura 4-5. Comparativa de eficiencia media



Nombre de la Fuente: Autor.

4.3 Resultados en sistema embebido

Como análisis final, se trasladó la implementación del algoritmo original y el algoritmo generado con esta investigación a un sistema embebido PYNQ-Z2. Algunas de las características principales de este sistema son:

- Procesador Cortex-A9 dual-core con 650MHz.
- Memoria 512MB DDR3 / 128Mbit FLASH
- Almacenamiento a través de puerto Micro SD.
- Equivalente a FPGA Artix-7 (1.3M puertos reconfigurables).
- Sistema operativo basado en Linux.
- Alimentación por USB o 7V-15V de fuente externa.
- Conexión Ethernet 10/100/1000 a través de Gigabit Ethernet PHY.

Este sistema embebido es programable con Python a través de la interfaz de Jupyter-Notebook, accesible por red local con el puerto Ethernet. Esto generó practicidad a la hora de trasladar los ensayos prácticos realizados en el equipo de cómputo ya que presentaba exactamente la misma interfaz.

36 Algoritmo para la clasificación en línea de fonemas de habla silenciosa utilizando un sistema embebido

Con las anteriores características, se reprodujo las pruebas prácticas, y se evidenciaron resultados equivalentes a los presentados anteriormente. En la tabla 4-2 se evidencia la información.

Sujeto	Media de clasificación [%]	Desviación estándar [%]	Tiempo etapa descomposición [s]	Tiempo etapa de caracterización [s]	Tiempo etapa de clasificación [s]	Tiempo total [s]
S1	0.87	0.05	225.98	20.25	8.46	254.69
S2	0.98	0.03	245.80	19.78	8.36	273.93
S3	0.74	0.11	240.57	20.14	8.65	269.35
S4	0.98	0.03	227.95	19.63	9.29	256.87
S5	0.94	0.06	227.41	21.14	8.37	256.92
S6	0.94	0.04	234.50	19.61	8.34	262.45
S7	0.97	0.03	251.97	20.22	8.57	280.76
S8	0.99	0.02	264.90	20.96	8.31	294.17
S9	0.95	0.04	270.09	20.84	8.37	299.31
S10	0.83	0.12	265.61	20.40	8.41	294.42
S11	0.96	0.05	248.18	20.13	8.27	276.58
S12	0.90	0.07	258.15	20.87	8.41	287.42
S13	0.94	0.05	286.38	20.40	7.74	314.52
S14	0.88	0.04	321.08	20.71	7.75	349.55
S15	0.93	0.07	231.84	20.19	8.53	260.57
S16	0.87	0.04	242.60	20.27	8.37	271.24
Media	0.92	0.05	252.69	20.35	8.39	281.42
Total			4043.00	325.56	134.18	4502.75

Tabla 4-2. Métricas obtenidas con el algoritmo generado en PYNQ

Observando los resultados de la Tabla 3-9 y la Tabla 4-1, se puede notar que el tiempo promedio de clasificación pasó de ser 13.59 segundos a 281.42 segundos, lo cual significa que el procesamiento de los datos en la PYNQ es casi 21 veces más lento que el tiempo de procesamiento empleado en el equipo de cómputo.

A pesar de que con el algoritmo generado empleó 281.42 segundos en procesar los triales de cada usuario en promedio, si tenemos en cuenta que con un porcentaje de superposición de 50% se tenía una base de datos de 175 triales por sujeto, se calcula entonces que el tiempo promedio empleado en la clasificación de cada trial es de 1.6 segundos en la tarjeta PYNQ-Z2.

4.4 Análisis de resultados

A continuación, se presenta una comparativa de resultados entre el punto de partida de esta tesis, y el resultado final del desarrollo de la misma, así como los aportes logrados específicamente por el autor.

Inicialmente el proceso completo de procesamiento y clasificación de la base de datos tardaba en promedio **2838.94 segundos** (ver Tabla 3-1) en dar resultados de clasificación de la muestra completa de triales por cada sujeto. Con el algoritmo generado en esta tesis se logró clasificar los datos equivalentes en un tiempo de procesamiento de **13.59 segundos** (ver Tabla 3-9). Los valores presentados anteriormente son los resultados de la clasificación en el **sistema de cómputo**, condiciones bajo las cuales se logró evidentemente una reducción sustancial del tiempo de procesamiento, siendo el tiempo inicial más de **200 veces** el tiempo logrado finalmente.

Con dicho logro, se procedió a la ejecución de los algoritmos iniciales y el algoritmo final en el sistema embebido con las características presentadas en el numeral 4.3 de este documento. La ejecución del algoritmo inicial basado en APIT-MEMD no presentaba resultados en el sistema embebido, el cuál resultaba bloqueado debido al alto costo computacional del algoritmo, lo que consumía la totalidad de recursos del hardware, llevándolo a un estado de bloqueo y haciendo inviable la ejecución de la clasificación en línea usando dicho algoritmo. Sin embargo, con el algoritmo generado en esta tesis, es

decir el basado en SVD-MEMD, **se logró la ejecución del algoritmo sin mayor inconveniente**, obteniendo los resultados de clasificación de los datos de cada sujeto en un tiempo promedio de **281.42 segundos** (ver Tabla 4-2), es decir aproximadamente **1.6 segundos** en procesar cada trial en el sistema embebido. Por lo anterior se evidencia el cumplimiento del objetivo general de esta tesis, el cual era *“Desarrollar un algoritmo para la clasificación en línea de fonemas del habla silenciosa, a partir del estudio y optimización de distintos factores de incidencia de la metodología basada en APIT-MEMD previamente desarrollada en el grupo de investigación CMUN, con el fin de ser implementado en un sistema embebido.”*.

Los valores presentados se lograron a través de un arduo trabajo de investigación, análisis y discretización de información referente a la naturaleza de las señales, la gran cantidad de formas de filtrado, caracterización y el abanico de opciones en clasificación, el cual fue ejecutado por el autor del documento. Presentando como aporte la generación del algoritmo basado SVD-MEMD, el cuál se destaca por sus resultados, gracias a la cohesión lograda entre las etapas de preprocesamiento, como la selección de frecuencia de muestreo, la discretización y superposición de las muestras, el cambio del algoritmo en la etapa de descomposición, la etapa de caracterización mejorada gracias a la aplicación del reductor de características, y la sintonización correcta de los clasificadores.

Con los resultados obtenidos de clasificación en el sistema embebido, es decir en promedio 92% de asertividad, con desviación estándar de 5% y un tiempo de procesamiento por trial de 1.6 segundos aproximadamente, es notorio que el ideal de un sistema de **procesamiento en tiempo real no está tan distante**. Sin embargo, en el punto actual de la investigación, no es viable la implementación del algoritmo como producto final. Esto teniendo en cuenta que, en la vida cotidiana de un usuario o paciente, la clasificación errónea de una orden puede conllevar a la ejecución de una acción indeseada que puede o no, llegar a poner en peligro a su usuario o la gente a su alrededor.

5. Conclusiones y recomendaciones

5.1 Conclusiones

Durante el proceso investigativo del proyecto, se encontraron varios factores sujetos de mejora, entre los cuales estuvo la frecuencia de muestreo ingresada a la etapa de procesamiento de datos, el algoritmo para el filtrado inicial de la información, los métodos de reducción de características, la selección y configuración del clasificador e incluso mejoras a nivel del código programado. Al ir abordando cada uno de estos factores, se hizo notorio que, en la mayoría de los casos, la configuración en búsqueda de la reducción del costo computacional en búsqueda de una reducción en el tiempo empleado conllevaba la decisión de sacrificar un porcentaje de precisión en la clasificación lograda con el algoritmo. Adicionalmente, la etapa en la cual realmente se obtuvieron los resultados más significativos fue la de descomposición basada en EMD; en esta etapa especialmente, se evidencia la necesidad de tener un conocimiento de la naturaleza de la señal tratada, y más puntualmente, conocimiento respecto a en que parte de la señal tratada se encuentra localizada la información de interés. Se menciona lo anterior como conclusión principal, debido a que el logro alcanzado se obtuvo gracias a reconocer el rango de frecuencias de interés al descomponer la información, y tomar la decisión de optar por el algoritmo SVDMEMD; esta decisión fue confusa, a causa que si se analizara solo la descomposición, tendríamos un resultado negativo, pues se generaron mezclas de modas dentro de las IMF extraídas, pero con la fortuna de que las modas mezcladas contenían la información dentro de los rangos de frecuencia de interés, se pudo utilizar dichas IMF como sustento para el resto del proceso de clasificación y obtener una mejora visible en la eficiencia.

5.2 Recomendaciones

Se sugiere que, en vista de generar portabilidad y adaptabilidad del hardware, se realice una búsqueda de herramientas cuyo software no presente mayores limitaciones y dependencias, permitiendo así, el uso de este bajo diferentes condiciones. Esta recomendación surge a partir de las dificultades presentadas con el equipo de hardware de G-Tec seleccionado para la adquisición de señales EEG, debido a que este equipo cuenta con unas limitaciones de software bastante puntuales a fecha del desarrollo de este proyecto, como por ejemplo la necesidad de Windows como sistema operativo, y una versión puntual de Matlab requerida (2015 a). Lo anterior hizo que finalmente no tuviera sentido continuar las pruebas con la base de datos obtenida con el equipo G-Tec, ya que finalmente no se podría implementar un algoritmo en línea con la información captada con dicho hardware. Se recomienda entonces, tener en cuenta las características y limitaciones de software en la selección de los equipos de hardware. Por otra parte, gracias a la experiencia obtenida en la investigación, se recomienda un estudio a profundidad de la naturaleza de los datos a estudiar, teniendo muy presente el objetivo a lograr y de esta manera concentrarse en extraer la información pertinente más que hacer un tratamiento perfecto de los datos. Finalmente se debe aclarar que a pesar de que se tengan múltiples herramientas, como en este caso, clasificadores, si se dedica algo de tiempo al sintonizado de estos recursos, se pueden obtener resultados incluso mejores de los ya alcanzados.

5.3 Continuidad

El proceso investigativo de este proyecto se llevó a cabo con el fin de acercarse a la implementación de un producto de prótesis robótica, el cual clasifique en tiempo real las señales EEG, y ejecute como movimientos en la prótesis las ordenes generadas por el usuario haciendo uso del habla silenciosa. Como se mencionó en el análisis de resultados, esto aún no se logra, sin embargo, como autor de esta tesis, pienso que es logable, y para darle continuidad se podría inicialmente hacer un cambio de lenguaje de programación, pasando de ser Python como lenguaje interpretado, a un lenguaje compilado como Java, o C, adicionalmente se puede diseñar un sistema embebido específicamente para la ejecución del algoritmo y el control de la prótesis. Con estos dos factores, muy seguramente se lograría otra disminución notoria en el tiempo de ejecución. Sin embargo,

quedaría al pendiente la determinación de a partir de cuál porcentaje de asertividad es responsable decir que los resultados son aplicables a la cotidianidad de un paciente.

Bibliografía

- [1] S. U. Arias, "Principales retos del Derecho Internacional Humanitario (DIH) en el contexto de postconflicto en Colombia," 2019.
- [2] "Víctimas de Minas Antipersonal y Municiones sin Explosionar." <http://www.accioncontraminas.gov.co/estadisticas/Paginas/victimas-minas-antipersonal.aspx> (accessed Feb. 10, 2020).
- [3] Kim. Norton, "A brief history of prosthetics," *InMotion*, vol. 17, no. 7, pp. 11-3-undefined, 2007.
- [4] S. Daniela García and V. María José Espinoza, "Avances en prótesis: una mirada al presente y al futuro," *Revista Médica Clínica Las Condes*, vol. 25, no. 2, pp. 281–285, Mar. 2014, doi: 10.1016/S0716-8640(14)70039-2.
- [5] "History | AOPA – AMERICAN ORTHOTIC & PROSTHETIC ASSOCIATION." <https://www.aopanet.org/about-aopa/history/> (accessed Jul. 11, 2021).
- [6] J. L. Brito, M. X. Quinde, D. Cuzco, and J. I. Calle, "Estudio del estado del arte de las prótesis de mano," 2013, Accessed: Jul. 11, 2021. [Online]. Available: <http://dspace.ups.edu.ec/handle/123456789/8447>
- [7] H. A. ROMO., J. C. REALPE, and P. E. JOJOA, "ANÁLISIS DE SEÑALES EMG SUPERFICIALES Y SU APLICACIÓN EN CONTROL DE PRÓTESIS DE MANO," *Avances en Sistemas e Informática*, vol. 4, no. 1, Jan. 2007, Accessed: Jul. 11, 2021. [Online]. Available: <https://revistas.unal.edu.co/index.php/avances/article/view/9725>
- [8] A. A. Adewuyi, L. J. Hargrove, and T. A. Kuiken, "Evaluating EMG Feature and Classifier Selection for Application to Partial-Hand Prosthesis Control," *Frontiers in Neurorobotics*, vol. 10, p. 15, Oct. 2016, doi: 10.3389/FNBOT.2016.00015.

- [9] E. Park and S. G. Meek, "Fatigue compensation of the electromyographic signal for prosthetic control and force estimation," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 10, 1993, doi: 10.1109/10.247800.
- [10] A. Kübler, V. K. Mushahwar, L. R. Hochberg, and J. P. Donoghue, "BCI Meeting 2005 - Workshop on clinical issues and applications," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 131–134, Jun. 2006, doi: 10.1109/TNSRE.2006.875585.
- [11] A.-B. Suleiman, T. A. Fathi, A.-B. R. Suleiman, and A.-H. Fatehi, "FEATURES EXTRACTION TECHNIQUES OF EEG SIGNAL FOR BCI APPLICATIONS", Accessed: Jul. 11, 2021. [Online]. Available: <https://www.researchgate.net/publication/228450475>
- [12] C. Park, D. Looney, N. ur Rehman, A. Ahrabian, and D. P. Mandic, "Classification of Motor Imagery BCI Using Multivariate Empirical Mode Decomposition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 1, Jan. 2013, doi: 10.1109/TNSRE.2012.2229296.
- [13] M. Rohm *et al.*, "Hybrid brain–computer interfaces and hybrid neuroprostheses for restoration of upper limb functions in individuals with high-level spinal cord injury," *Artificial Intelligence in Medicine*, vol. 59, no. 2, Oct. 2013, doi: 10.1016/j.artmed.2013.07.004.
- [14] F. Cincotti *et al.*, "Non-invasive brain–computer interface system: Towards its application as assistive technology," *Brain Research Bulletin*, vol. 75, no. 6, Apr. 2008, doi: 10.1016/j.brainresbull.2008.01.007.
- [15] J. H. Friedman, "Multivariate Adaptive Regression Splines," <https://doi.org/10.1214/aos/1176347963>, vol. 19, no. 1, pp. 1–67, Mar. 1991, doi: 10.1214/AOS/1176347963.

- [16] N. Rehman and D. P. Mandic, "Multivariate empirical mode decomposition," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 466, no. 2117, May 2010, doi: 10.1098/rspa.2009.0502.
- [17] A. Hemakom, V. Goverdovsky, D. Looney, and D. P. Mandic, "Adaptive-projection intrinsically transformed multivariate empirical mode decomposition in cooperative brain-computer interface applications," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, Apr. 2016, doi: 10.1098/rsta.2015.0199.
- [18] S. Zhao and F. Rudzicz, "Classifying phonological categories in imagined and articulated speech," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-August, pp. 992–996, Aug. 2015, doi: 10.1109/ICASSP.2015.7178118.
- [19] S. I. Villamizar Delgado, "Development of algorithms to improve the technical efficiency of capturing, processing, and identification of EEG signals in the word imagery task," Feb. 2020, Accessed: Jul. 11, 2021. [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/77829>
- [20] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998, doi: 10.1098/RSPA.1998.0193.
- [21] Z. Wu and N. E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046, pp. 1597–1611, Jun. 2004, doi: 10.1098/RSPA.2003.1221.
- [22] G. Rilling, P. Flandrin, and P. Gonçalvès, "ON EMPIRICAL MODE DECOMPOSITION AND ITS ALGORITHMS".
- [23] G. Rilling, P. Flandrin, P. Goncalves, and J. M. Lilly, "Bivariate empirical mode decomposition," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 936–939, Dec. 2007, doi: 10.1109/LSP.2007.904710.

- [24] N. Ur Rehman and D. P. Mandic, "Empirical mode decomposition for trivariate signals," *IEEE Transactions on Signal Processing*, vol. 58, no. 3 PART 1, pp. 1059–1068, Mar. 2010, doi: 10.1109/TSP.2009.2033730.
- [25] J. Fleureau, A. Kachenoura, L. Albera, J. C. Nunes, and L. Senhadji, "Multivariate empirical mode decomposition and application to multichannel filtering," *Signal Processing*, vol. 91, no. 12, pp. 2783–2792, Dec. 2011, doi: 10.1016/J.SIGPRO.2011.01.018.
- [26] X. Lang *et al.*, "Fast Multivariate Empirical Mode Decomposition," *IEEE Access*, vol. 6, pp. 65521–65538, 2018, doi: 10.1109/ACCESS.2018.2877150.
- [27] H. C. Andrews and C. L. Patterson, "Singular Value Decomposition (SVD) Image Coding," *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 425–432, 1976, doi: 10.1109/TCOM.1976.1093309.
- [28] C. Moore, "APPLICATION OF SINGULAR VALUE DECOMPOSITION TO THE DESIGN, ANALYSIS, AND CONTROL OF INDUSTRIAL PROCESSES," *Proceedings of the American Control Conference*, pp. 643–650, 1986, doi: 10.23919/ACC.1986.4789019.
- [29] R. Izmailov, D. Bassu, A. McIntosh, L. Ness, and D. Shallcross, "Application of multi-scale singular vector decomposition to vessel classification in overhead satellite imagery," *Seventh International Conference on Digital Image Processing (ICDIP 2015)*, vol. 9631, p. 963108, Jul. 2015, doi: 10.1117/12.2196925.
- [30] "Understanding Singular Value Decomposition and its Application in Data Science | by Reza Bagheri | Towards Data Science." <https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d> (accessed Jul. 11, 2021).
- [31] "Una introducción a los Árboles de Decisión | DABIA." <https://www.grupodabia.com/post/2020-05-19-arbol-de-decision/> (accessed Jul. 11, 2021).

- [32] “6 AdaBoost | Modelos Predictivos.” https://fhernanb.github.io/libro_mod_pred/adaboost.html (accessed Jul. 11, 2021).
- [33] M. Mandal and A. Mukhopadhyay, “An Improved Minimum Redundancy Maximum Relevance Approach for Feature Selection in Gene Expression Data,” *Procedia Technology*, vol. 10, pp. 20–27, Jan. 2013, doi: 10.1016/J.PROTCY.2013.12.332.
- [34] S. Ramírez-Gallego *et al.*, “Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data,” *International Journal of Intelligent Systems*, vol. 32, no. 2, pp. 134–152, Feb. 2017, doi: 10.1002/INT.21833.
- [35] P. Bugata and P. Drotar, “On some aspects of minimum redundancy maximum relevance feature selection,” *Science China Information Sciences 2019 63:1*, vol. 63, no. 1, pp. 1–15, Dec. 2019, doi: 10.1007/S11432-019-2633-Y.
- [36] A. Atyabi, S. Fitzgibbon, and D. Powers, “The impact of Biasing on Overlapping windows : An EEG study,” Jul. 2012.
- [37] A. Atyabi, S. P. Fitzgibbon, and D. M. W. Powers, “Multiplication of EEG Samples through Replicating, Biasing, and Overlapping,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7670 LNAI, pp. 209–219, 2012, doi: 10.1007/978-3-642-35139-6_20.
- [38] A. Atyabi, S. Fitzgibbon, and D. M. W. Powers, “Biasing the overlapping and non-overlapping sub-windows of EEG recording,” Jun. 2012. doi: 10.1109/IJCNN.2012.6252465.