EEG Signal Clustering With Learned Features Using Deep Autoencoder

Sergio Villazana *,[a], César Seijas [a], Guillermo Montilla [b], Egilda Pérez [a]

[a] *Universidad de Carabobo, Facultad de Ingeniería, Centro de Procesamiento de Imágenes, Valencia, Venezuela*
[b] *Yttrium-Technology Corp., Panamá, Panamá.*

La selección de este artículo fue realizada en el marco de la Jornada de Investigación de la Escuela de Ingeniería Eléctrica "Prof. César Rodolfo Ruiz" Octubre 2020, siendo la evaluación, arbitraje, aceptación y edición a cargo de Revista Ingeniería UC.

https://doi.org/10.54139/revinguc.v28i1.18

Abstract.- This work proposes a convolutional autoencoder based non-supervised feature extractor, to find clusters of electroencephalographic signal (EEG) to supporting to the physicists to diagnose the epilepsy condition. Three autoencoders were designed with input dimensions of 4096×1, 2048×2 and 768×6, to analyze the impact of the signal length on latent representation generated by autoencoders. Latent representation was used as input to the clustering algorithms *K*-means and support vector clustering. Latent representation was mapped onto a two-dimensional space of mean and standard deviation to visualize it, and to apply the clustering algorithms. Results showed a good latent representation of the three autoencoders, with a maximum reconstruction error of 1,47 % for the worst case. Clustering algorithms got visually consistent clusters compared with the ground-truth distribution onto the two-dimension latent space. The best performance was achieved with the *K*-means algorithm and the best latent representation of the input signal. Resultant clusters were impacted by the length of the input segment, where *K*-means with an input length of 4096 samples had the best performance.

Keywords: Epileptic EEG signals; autoencoder; *K*-means; SVC.

Agrupamiento de Señales EEG con Rasgos Aprendidos Usando Autoencoder Profundo

Resumen.- Este trabajo propone un algoritmo basado en autoencoders convolucionales como extractor de rasgos no supervisado, para hallar grupos o clusters de señales electroencefalográficas (EEG), como apoyo para el especialista médico para facilitar el diagnóstico de la condición de epilesia. Se diseñaron tres autoencoders con señales de entrada de 4096×1, 2048×2 y 768×6, para analizar el efecto de la longitud de la señal sobre la representación latente generada por los autoencoders. Las representación latente se utilizó como entrada a los algoritmos de agrupamiento *K*-means y basado en vectores de soporte. La representación latente se llevó a un espacio bidimensional donde se obtuvo la media y la desviación estándar para visualizarla, y operar sobre ellas los algoritmos de agrupamiento. Los resultados demostraron una buena representación latente de los tres autoencoders, con un error máximo de reconstrucción de las señales de entrada de 1,47 % para el peor caso. Los algoritmos de agrupamiento lograron obtener unos grupos visualmente consistentes con la distribución de los puntos de referencia en el espacio bidimensional latente. La mejor medida de desempeño se logró con el algoritmo *K*-means con la mejor representación latente de las señales de entrada. Los grupos resultantes fueron influenciados por la longitud del segmento de entrada, donde el algoritmo *K*-means con una longitud de entrada de 4096 muestras tuvo la mejor medida de desempeño.

Palabras clave: Señales EEG epilépticas; autoencoders; *K*-means; agrupamiento basado en vectores de soporte.

Recibido: 22 de octubre, 2020.
Aceptado: 27 de noviembre, 2020.

1. Introducción

Uno de los desórdenes neurológicos más importante que afecta a la actividad cerebral es la epilepsia. La epilepsia es una condición que padecen 50 millones de personas alrededor del mundo [1], que produce crisis convulsivas que afecta la calidad de vida del paciente [2]. Para la detección de la condición de epilepsia el especialista tiene que analizar e interpretar un conjunto de señales electroencefalográficas (EEG) muy extensas, lo que hace que la detección sea un proceso muy tedioso, y además dependiente

* Autor para correspondencia:
*Correo-e:*svillaza@gmail.com (S. Villazana)

del criterio del especialista que realiza dicho análisis. Es imperativo automatizar la detección de señales epilépticas mediante el análisis de las señales EEG para coadyuvar al especialista en su tarea de analizar estas señales. Actualmente, existen técnicas para el análisis de las señales EEG, basados en algoritmos de inteligencia artificial, destacando en los últimos años las redes neuronales profundas (entre ellas las redes convolucionales y las redes recurrentes). Las diferentes clases de señales epilépticas poseen rasgos subyacentes particulares que caracterizan a cada clase en consideración, pero los mismos no son visibles en las señales EEG temporales. Estos rasgos subyacentes permiten discriminar entre diferentes señales EEG normales y patológicas, y por tanto, asignarles una etiqueta; este proceso de análisis se denomina *clustering* (agrupamiento) [3]. El *clustering* es una técnica de análisis de datos que permite asignarles etiqueta a los datos, lo que se puede entender como una técnica de clasificación no supervisada [3], ya que no se cuenta con etiquetas previas con las cuales compararlas.

Una característica a destacar de las redes neuronales profundas es la capacidad de extraer la información subyacente (rasgos) de los datos de entrada, por medio de sus numerosas capas intermedias, lo que elimina la necesidad utilizar enfoques para extraer rasgos a mano, que utilizan una vasta cantidad de técnicas lineales y no lineales, basadas en el tiempo o en la frecuencia o combinación de ellas. La efectividad de los rasgos extraídos manualmente de la señal para modelar exitosamente los rasgos subyacentes, y detectar los grupos (*clusters*) "naturales" de los datos es dependiente de la naturaleza de los datos bajo análisis, y por tanto, exige un trabajo arduo del investigador para la selección de los mismos durante la fase de experimentación.

Los algoritmos de aprendizaje profundo para extraer los rasgos subyacentes de las señales EEG, cuando no se cuenta con una discriminación a priori (etiqueta o clase) de las señales, son los *autoencoders* (AE) [4]. Los AE son redes neuronales convolucionales profundas que, fundamentalmente, tratan de emular la función identidad, es decir, la salida es una reproducción de su entrada [5]. Los AE se componen de dos bloques en cascada denominados codificador (*encoder*) y decodificador (*decoder*). La salida del primero es una representación latente (rasgo subyacente) de la entrada, que es la entrada del decodificador, que finalmente reproduce la entrada [5]. En la medida que el error de reconstrucción (diferencia de aproximación entre la entrada y la salida) sea más pequeño, la representación latente contiene rasgos más relevantes de la entrada [5].

Los métodos para agrupar los datos, en este trabajo, con apoyo en la representación latente (rasgos subyacentes) obtenidos con el AE son el agrupamiento basado en *K*-means [6] y agrupamiento basado en vectores de soporte (*Support Vector Clustering*, SVC de sus siglas en inglés) [7, 8]. En el agrupamiento *K*-means [6] conocido también como agrupación dura (*hard clustering*), los datos se dividen en grupos distintos, donde cada elemento de dato pertenece exactamente a un grupo (*cluster*). El agrupamiento basado en las SVC es un algoritmo que puede detectar formas arbitrarias de grupos de un espacio de datos [7]. EL SVC proyecta los puntos de datos a un espacio de rasgos hiperdimensional usando una función de núcleo (*kernel*) Gaussiana [8].

Este estudio propone un proceso de extracción no supervisada de rasgos señales EEG usando autoencoders convolucionales, para agrupar las señales EEG en grupos (clusters) utilizando dos métodos de agrupamiento basado en *K*-means y SVC.

2. Fundamentación teórica

2.1. Autoencoders

Los autoencoders son redes neuronales profundas para obtener una representación de los datos en un nuevo espacio conocido como espacio latente. El autoencoder está compuesto de dos bloques fundamentales: a) El encoder que transforma los datos de entrada, x, a un espacio, de menor dimensión que el espacio original de los datos, mientras que mantiene los rasgos más relevantes $z = f_\phi(x)$ [5], donde f y ϕ son la función de proyección del espacio de entrada al espacio latente, y los parámetros de la función de

proyección f del encoder, respectivamente; b) El decoder que reconstruye los datos originales a partir de la representación latente, $r = g_\theta(z) \cong x$ [5], donde g y θ son la función de proyección del espacio latente (z) al espacio de entrada, y los parámetros de la función de proyección g del decoder, respectivamente. El AE sintetiza una composición de funciones $r = g_\theta(f_\phi(x)) \cong x$, donde para obtener una buena representación latente (z) de la entrada (x), se requiere que la reconstrucción r sea lo más aproximado posible a la entrada x. La función objetivo a minimizar, para obtener la mejor reconstrucción de la entrada, es el error medio cuadrático. Dado el conjunto de entrada $x_1, x_2, \ldots, x_n$, se requiere minimizar la función objetivo $\min_{\phi,\theta} \frac{1}{n} \sum_{i=1}^{n} \|x_i - g_\theta(f_\phi(x_i))\|^2$, donde ϕ y θ son los parámetros del encoder y el decoder, respectivamente [5].

2.2. *Agrupamiento basado en las Máquinas de Vectores de Soporte*

El agrupamiento basado en las Máquinas de Vectores de Soporte (SVC siglas en inglés de ***S**upport **V**ector **C**lustering*) se basa sobre el clasificador de una clase [9, 10] y un enfoque de etiquetamiento de los grupos [8]. El objetivo principal de esta máquina de aprendizaje es hallar al hiperplano más alejado del origen, en el espacio de rasgos, que separa a todos los datos hacia el lado (del hiperplano) opuesto al origen. Esto se logra al minimizar la función objetivo $\min \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho$, sujeto a las restricciones $\mathbf{w}^T \Phi(x_i) \geq \rho - \xi_i$, $\xi_i \geq 0$, $\rho \geq 0$, donde $\mathbf{w}$ es la normal del hiperplano, ρ es la distancia del hiperplano al origen, y ν controla la influencia de los datos atípicos (*outliers*). Φ es una función no lineal (función kernel) que proyectan los puntos desde el espacio de datos a un espacio de rasgos hiperdimensional. La función Langragiano de la función objetivo es $L = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho - \sum_{i=1}^{n} \alpha_i(\mathbf{w}^T \Phi(x_i) + \xi_i - \rho) - \sum_{i=1}^{n} \beta_i \xi_i$, sujeta a α_i, $\beta_i \geq 0$. Tomando las derivadas del Langragiano L con respecto a $\mathbf{w}$, ξ_i, ρ e igualando a cero conduce a $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$, $\alpha_i + \beta_i = 1/\nu n$, $\sum_{i=1}^{n} \alpha_i = 1$. El producto interno entre todos los vectores $\Phi(x_i)$ y $\Phi(x_j)$ en el espacio de rasgos es la matriz kernel, y un elemento de dicha matriz es $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$. La función kernel a utilizar es la función Gaussiana, $K(x_i, x_j) = \mathrm{e}^{(-\|x_i - x_j\|^2/(2\sigma^2))}$, como lo propuso [8], donde el parámetro σ es el ancho de la Gaussiana, algunos autores utilizan la siguiente igualdad $\gamma = 1/2\sigma^2$. Sustituyendo las relaciones obtenidas del Langragiano y sustituyéndolas en la función objetivo primaria conduce a la forma dual de esta $\min \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle = \min \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$. Las condiciones KKT [11] establecen que $\alpha_i(\mathbf{w}^T \Phi(x_i) + \xi_i - \rho) = 0$ y $\beta_i \xi_i = (1/\nu n - \alpha_i) = 0$. Los vectores de soporte (VS) son aquellos datos x_i para los cuales los multiplicadores de Lagrange α_i son distintos de cero con $\xi_i \geq 0$, y los datos x_i ordinarios que no son VS su correspondiente α_i son iguales a cero.

El enfoque de etiquetamiento o de asignación de grupos se basa en una técnica muy rápida propuesta en [8], este enfoque de etiquetamiento de grupo está basado en conos [8]. Debido a que la función kernel es la Gaussiana, se puede demostrar, que los datos en el espacio de entrada se proyectan a una hiperesfera de radio unitario, de allí que los autores definen un cono entre dos vectores de soporte. Si dos o más conos se interceptan los vectores de soporte que definen a cada cono pertenecen al mismo grupo. En este caso, el máximo número de grupos corresponde al número de vectores de soporte. Cada cono se define por el ángulo que forman dos vectores de soporte y el origen. En [8] los autores demostraron que el ángulo entre las imágenes de dos vectores $\Phi(x_i)$ y $\Phi(x_j)$ en el espacio de rasgos, cuando la función kernel es una Gaussiana, está definida por $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \cos\omega$, donde ω es el ángulo entre los vectores $\Phi(x_i), \Phi(x_j)$. Se puede demostrar que las distancias en el espacio de datos corresponden a ángulos en el espacio de rasgos y viceversa. Esta afirmación es importante debido a que el proceso de agrupamiento es hecho en el espacio de datos usando la distancia, afectada por un factor arbitrario a criterio del investigador para

ajustar el número de grupos.

2.3. Agrupamiento basado en K-means

Este algoritmo divide a los datos en K grupos o particiones, $C_1, C_2, \ldots, C_K$, representados por sus centros. El centro de cada grupo se calcula como la media de todos datos que pertenecen a dicho grupo. Los K centros iniciales se seleccionan aleatoriamente. En cada iteración se reasignan los datos al grupo más cercano utilizando la distancia Euclidiana entre el k-ésimo centro y el dato bajo consideración. Luego los centros de los grupos se recalculan con los nuevos miembros incorporados y/o desincorporados. El k-ésimo centro se calcula como $\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$ [12], donde N_k es el número de datos que pertenecen al k-ésimo grupo y μ_k es la media del grupo k. El criterio de parada del algoritmo es observar el error de asignación al grupo, si este no se reduce con la reubicación de los centros, o alcanzar un número máximo de iteraciones, lo que suceda primero.

2.4. Medidas de evaluación de desempeño

Para evaluar el desempeño de los modelos de agrupamiento existen dos medidas ampliamente usadas en la literatura. La primera medida es la exactitud basada en el algoritmo Húngaro y la segunda es la información mutua normalizada.

2.4.1. Exactitud basada en el algoritmo Húngaro:

Es una medida directa que refleja la efectividad del algoritmo de agrupamiento. La exactitud viene dada por $ExacHung = \frac{1}{n} \sum_{i=1}^{n} \delta(c_i, map(\hat{c}_i))$, donde c_i es la etiqueta verdadera de los datos, y $\hat{c}_i$ es la etiqueta asignada por el algoritmo de agrupamiento; $\delta(i, j) = 1$ si $i = j$, $\delta(i, j) = 0$ si $i \neq j$; map es una función de proyección optimizada que proyecta cada cluster hallado a la clase real [13], y n es el número de datos. Esta medida halla la mejor correspondencia entre los grupos hallados y las clases de referencia [13].

2.4.2. Información mutua normalizada:

La información mutua normalizada (IMN) [14] se calcula como $IMN = 2 \times I(C, \hat{C}/(H(C) + H(\hat{C}))$, donde C son las clases de referencia, $\hat{C}$ son las etiquetas asignadas por el algoritmo de agrupamiento, I es la información mutua, y H es la entropía.

3. Metodología

3.1. Base de datos

La base de datos utilizada en este trabajo corresponde a las señales EEG de la Universidad de Bonn descrita en [15], está conformada por cinco conjuntos A, B, C, D y E, cada uno con 100 señales electroencefalográficas (EEG) monocanal de 23,6 segundos de duración cada una. Estas señales EEG están libres de artefactos debido a la actividad muscular o movimientos de los ojos. Los conjuntos A y B consisten en segmentos tomados de los registros EEG superficiales obtenidos de cinco voluntarios sanos usando un esquema de colocación de los electrodos estandarizada, conocida como sistema 10-20. Los conjuntos A y B corresponden a voluntarios despiertos, relajados y con los ojos abiertos (A) y los ojos cerrados (B). Los conjuntos C, D y E, de pacientes diagnosticados con epilepsia, corresponden a EEG profundos o intracraneales. Las señales en el conjunto C fueron obtenidas de la formación hipocampal del cerebro. El conjunto D se obtuvo dentro de la zona epileptogénica. Los conjuntos C y D contienen registros de la actividad cerebral medida durante los intervalos sin crisis epilépticas (interictal). El conjunto E contiene registros durante la actividad convulsiva, o periodo ictal.

Todos estos segmentos EEG se registraron con un amplificador de 128 canales, un convertidor analógico-digital de 12 bits a una frecuencia de muestreo de 173,61 Hz, y se les aplicó un filtro pasabanda con ajustes de 0,53 Hz y 40 Hz [15]. El número de registros en total es 500, con 4097 muestras cada uno (23,6 segundos de registro por EEG). La Figura 1 presenta una muestra de cada una de las cinco señales por cada conjunto, la unidad de los ejes verticales está en microvoltios (μV), y la unidad del eje horizontal es muestras (adimensional).

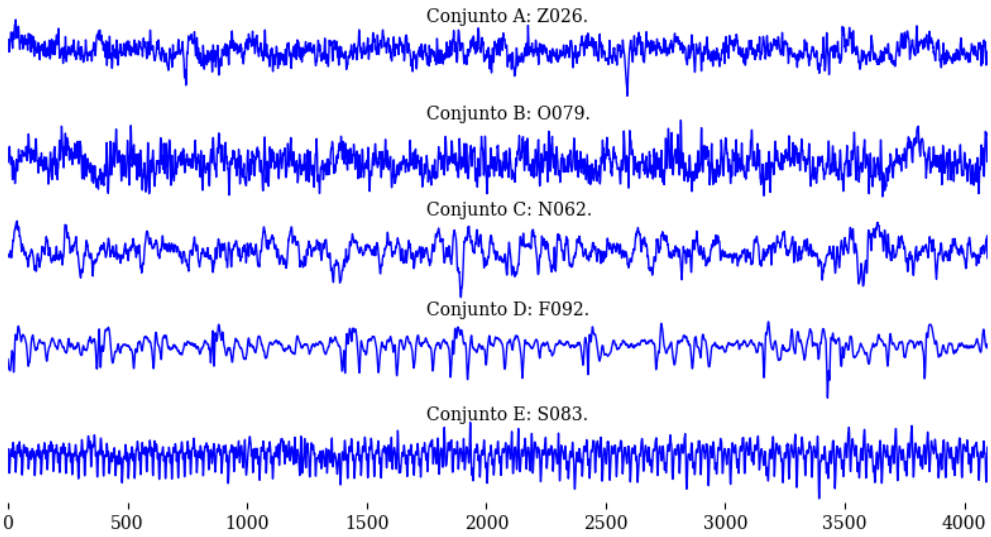

Figura 1: Muestras de señales de los conjuntos A, B, C, D y E

3.2. Entrenamiento de los autoencoders

La Figura 2 muestra un diagrama de bloques del proceso de extracción de rasgos (z) y de clustering. El proceso de entrenamiento está dividido en dos fases: a) Entrenamiento del autoencoder para obtener la representación latente de la señal (z) y, b) Entrenamiento del algoritmo de agrupamiento para determinar los grupos o clusters.

Para agrupar los datos se realizaron tres esquemas para la extracción de los rasgos con el autoencoder y analizar su efecto sobre los grupos generados. El primer esquema fue utilizar las señales unidimensionales completas (4096 muestras) como entrada al autoencoder, la segunda estrategia fue dividir cada señal en dos segmentos de 2048 muestras (descartando la última muestra) y formando con ellos una nueva señal de 2048×2, la tercera y última estrategia fue dividir cada señal en segmentos de 768 muestras. En esta última estrategia el número de dimensión de la nueva señal sería de 768×5, quedando descartadas 257 muestras, por lo que se decidió incluir un segmento adicional, tomando las últimas 511 muestras del quinto segmento, por tanto la dimensión resultante fue de 768×6.

La arquitectura del autoencoder para las señales de dimensión 4096×1 se muestra en la Tabla 1. Todas las capas de connvolución (Conv1D, Conv1DTranspose) se implementaron con un filtro de longitud 7 y el número de filtros (Profundidad) depende de la ubicación de la capa en la arquitectura de la red. La capa Conv1DTranspose (llamada también como capa de des-convolución), realiza la operación de convolución, pero, aumenta a su salida la dimensión de la señal de entrada, a diferencia de la capa Conv1D que reduce a su salida la dimensión de la señal de entrada. Todas las capas de Dropout eliminan aleatoriamente el 20 % de las neuronas de la capa precedente, estas se incluyeron para agregar un factor de regularización para mejorar la capacidad de generalización de la red. Todas las capas de convolución (Conv1D, Conv1DTranspose), excepto la última, le sigue una capa con la función de activación LeakyReLU con su parámetro por defecto (alpha = 0,3). El encoder lo conforman las capas de convolución, su entrada es la señal a codificar (4096×1) y su salida es de dimensión 1024×16, como se observa, se redujo su longitud de 4096 a 1024, pero se incrementó la dimensión (de 1 a 16). El decoder lo conforman las capas de des-convolución, con una entrada de 1024 × 16 y salida de 4096 × 1. El número de parámetros del autoencoder con señal de entrada de 4096×1 es 9505.

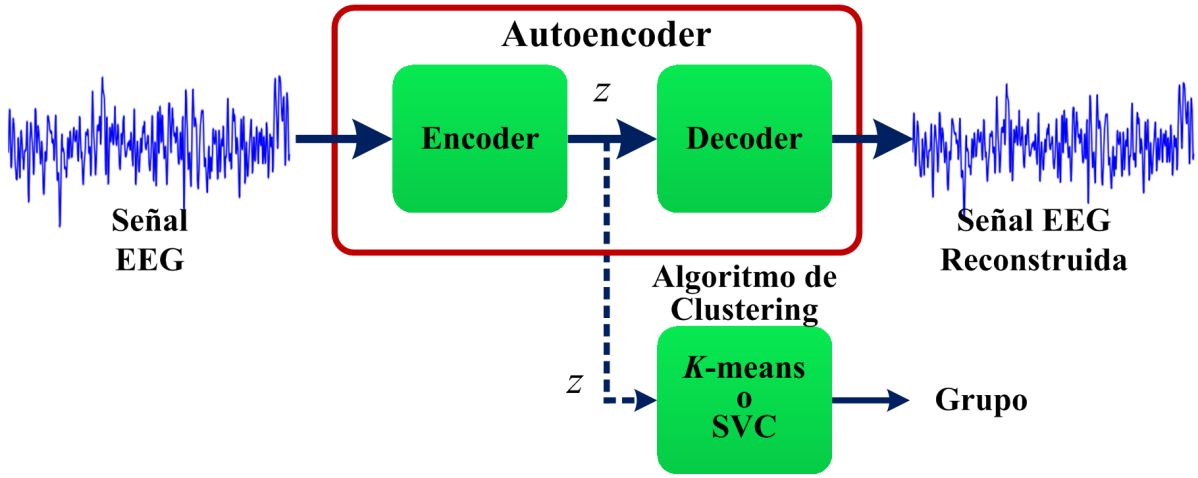

Figura 2: Diagrama de bloques del proceso de agrupamiento

Tabla 1: Arquitectura del autoencoder

	Tipo de capa	Filtro			Salida
		Dimensión	Profundidad	Paso	Dimensión
Encoder	Entrada	4096×1	-	-	-
	Conv1D	7×1	32	2	2048×32
	Dropout	-	-	-	2048×32
	Conv1D	7×1	16	2	1024×16
Decoder	Conv1DTranspose	7×1	16	2	2048×16
	Dropout	-	-	-	2048×16
	Conv1DTranspose	7×1	32	2	4096×32
	Conv1DTranspose	7×1	1	2	4096×1

La arquitectura de los autoencoders se mantiene para las otras configuraciones de entrada, variando solamente las dimensiones de entrada, la salida del autoencoder y las variables latentes (salida del encoder). El número de parámetros de los autoencoders para las entradas de 2048×2 y 768×6 es de 9954 y 11750, respectivamente, y las dimensiones de la salida del encoder son 512×16 y 192×16, para las señales de entrada de 2048×2 y 768×6, respectivamente.

Los ajustes para entrenar fueron: El optimizador elegido fue *Adam* con sus parámetros por defecto, la función objetivo fue *mse* (error medio cuadrático). El número de épocas fue 100, y el tamaño del lote (*batch size*) fue de 81.

La división de los datos, para todos los ensayos, fue de 90 % (450 señales) para la fase de entrenamiento y 10 % (50 señales) para prueba. Del conjunto de entrenamiento se tomó el 90 % (405 señales) para entrenar y 10 % (45 señales) para validación en cada época de entrenamiento, y supervisar si se está sobreentrenando (*overfitting*) o subentrenando (*underfitting*). Todas las señales de entrenamiento fueron escaladas por columnas entre 0 y 1, ambos valores inclusive. Las señales de prueba se escalaron entre 0 y 1 utilizando como referencia los valores máximos y mínimos de cada columna o dimensión de las señales de entrenamiento.

La Tabla 2 muestra los errores de entrenamiento y prueba (o generalización) de los autoencoders. Se observa un aumento de los errores en la medida que el número de muestras de la señal de entrada disminuye, ya que el número de parámetros se incrementa, lo que significa que la complejidad aumenta, y a mayor complejidad mayor error.

Tabla 2: Errores de entrenamiento y validación

Autoencoder		Errores (%)	
Entrada	Nro. de Parámetros	Entrenamiento	Prueba
4096×1	9505	0,033	0,220
2048×2	9954	0,117	0,604
768×6	11750	0,314	1,468

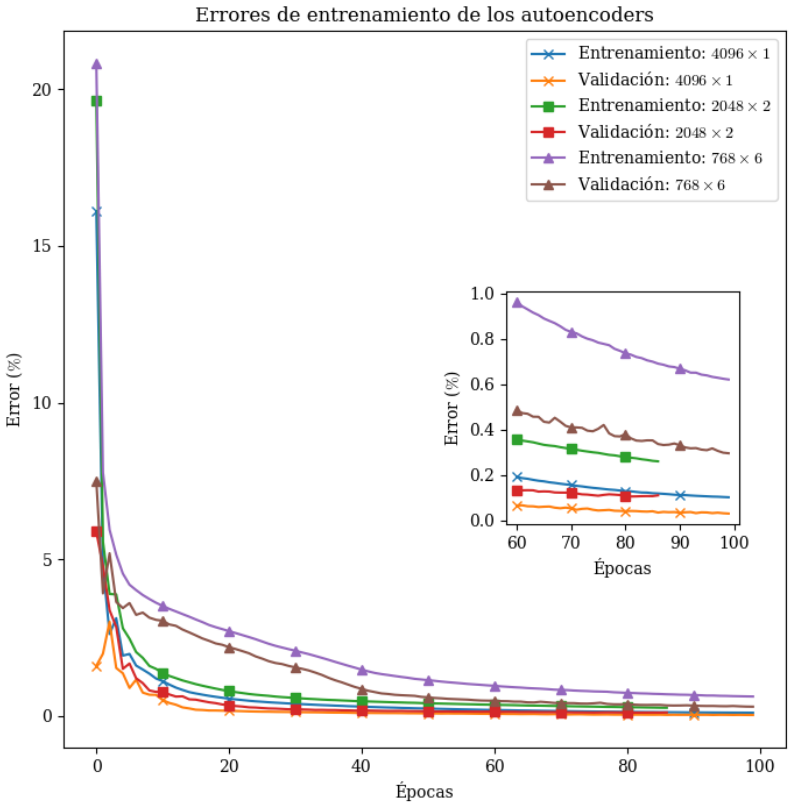

Figura 3: Errores de entrenamiento y validación durante el entrenamiento

La Figura 3 muestra los errores de entrenamiento y de validación en función de las épocas de entrenamiento de los tres autoencoders.

Se observa que los errores de validación siempre son menores que los errores de entrenamiento durante todo el entrenamiento, lo que significa que no hubo sobreentrenamiento de los autoencoders.

Las Figuras 4, 5 y 6 muestran las señales de entrada y salida (reconstrucción) de los encoders con entradas 4096×1, 2048×2 y 768×6, respectivamente.

Las Figuras 4, 5 y 6 muestran una excelente reconstrucción de las señales, lo que es un indicativo de que la representación latente (salida del encoder) de las entradas en bastante representativa de cada señal.

3.3. Entrenamiento de los algoritmos de clustering

El procedimiento para entrenar los algoritmos de agrupamiento fue el siguiente: Se tomaron las representaciones latentes de las señales de prueba y se redimensionaron como vectores unidimensionales, ello se hizo porque las dimensiones de la representación latente fueron de 1024×16, 512×16 y 192×16 para los autoencoders con entradas 4096×1, 2048×2 y 768×6, respectivamente. El número de vectores unidimensionales fue 50 con longitudes 16384, 8192 y 3072 para los autoencoders con entradas 4096×1, 2048×2 y 768×6, respectivamente. Luego por cada vector se obtuvo su media y su desviación estándar para obtener 50 vectores de dos dimensiones, $z = (z_0, z_1)$, donde z_0 es la media del vector unidimensional latente, y z_1 es la desviación estándar del vector unidimensional latente, para cada autoencoder. Las Figuras 7, 8 y 9 muestran la

Tabla 3: Parámetros del algoritmo SVC

Encoder	Parámetros del clasificador de una clase		Parámetro de etiquetamiento	Nro. de vectores de soporte	Número de grupos
Entrada	nu (ν)	sigma (σ)	Factor		
4096×1	0,38	0,0707	0,32	19	4
2048×2	0,30	0,0707	0,50	16	4
768×6	0,25	0,0707	0,45	15	5

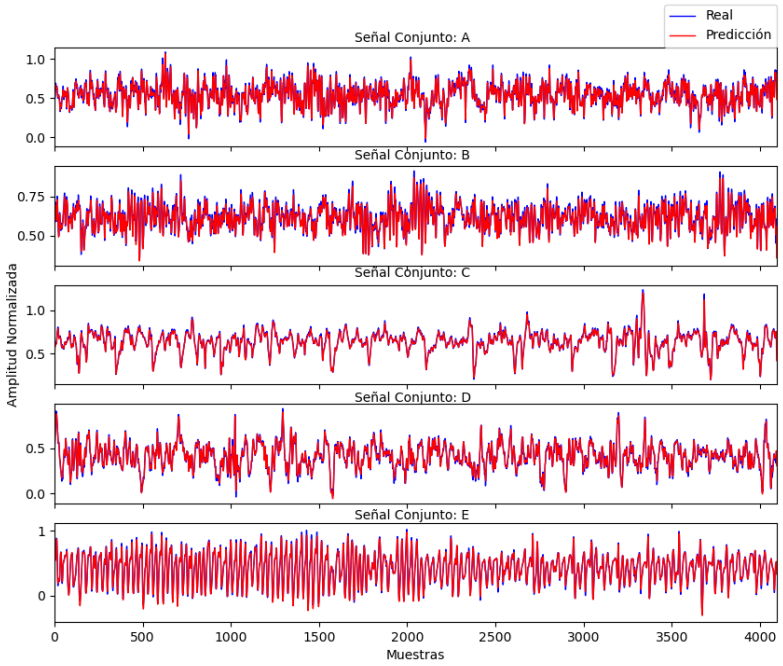

Figura 4: Señales de entrada de prueba y sus reconstrucciones. Encoder 4096×1.

distribución de los nuevos vectores latentes, donde también se indica el conjunto de origen de las señales EEG. Las distribuciones mostradas en las Figuras 7, 8 y 9 son las referencias (*ground-truth*) para evaluar el desempeño de los algoritmos de agrupamiento para todos los autoencoders.

Si se detallan cuidadosamente los rangos de los ejes de las Figuras 7, 8 y 9, se observa que las medias (eje horizontal) y las desviaciones estándar (eje vertical) de las variables latentes, independientemente de las clases, disminuyen con la longitud de la señal de entrada del encoder. También se ve que la posición relativa de los puntos, representando cada señal del conjunto de prueba, se mantienen aproximadamente invariable, independientemente del encoder. También es notable la superposición de los puntos pertenecientes a las señales de los conjuntos A y B (ambos de EEG normales). En general, el conjunto D es el que está bien diferenciado del resto de los conjuntos, algunos puntos del conjunto C se superponen con puntos del conjunto A, y el conjunto E (señales ictales), posee algunas señales que se solapan con los puntos de los conjuntos A y B, y otras señales que definitivamente se alejan muy claramente del resto.

3.3.1. Agrupamiento con el método K*-means*

Se fijó el número de clusters (grupos) igual al número de conjuntos de la base de datos, $K = 5$. Los centroides de los grupos se

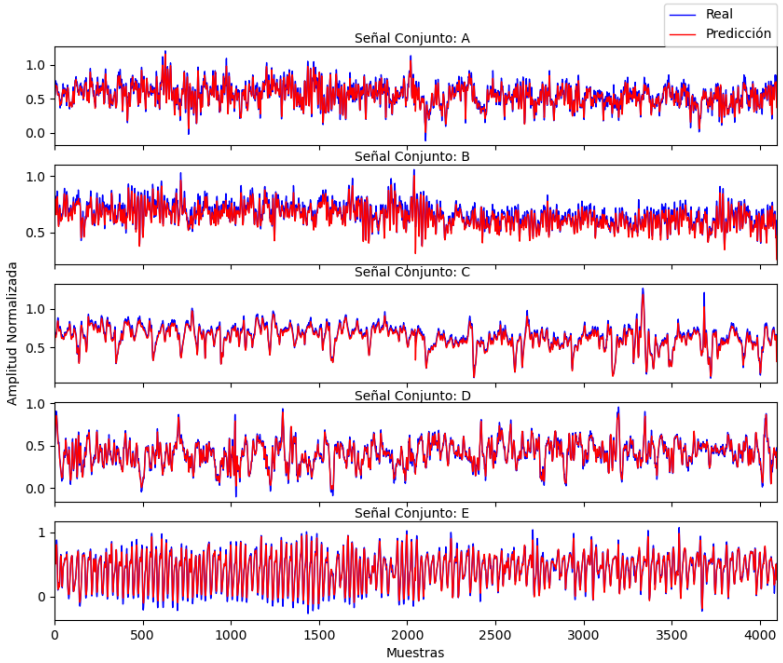

Figura 5: Señales de entrada de prueba y sus reconstrucciones. Encoder 2048×2.

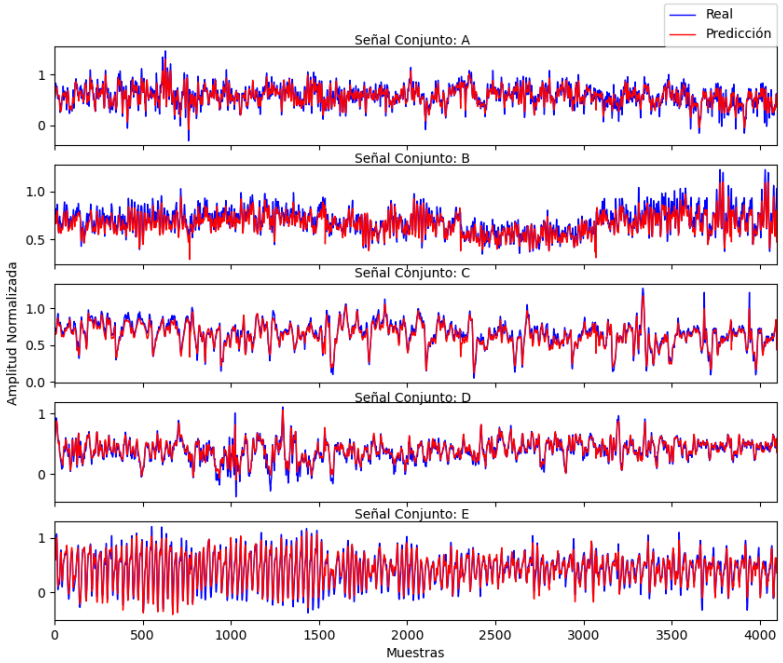

Figura 6: Señales de entrada de prueba y sus reconstrucciones. Encoder 768×6.

inicializaron aleatoriamente. El resultado para los tres autoencoders se muestran en la Figuras 10, 11 y 12. Se observa que la posición de los clusters con la misma referencia numérica no necesariamente

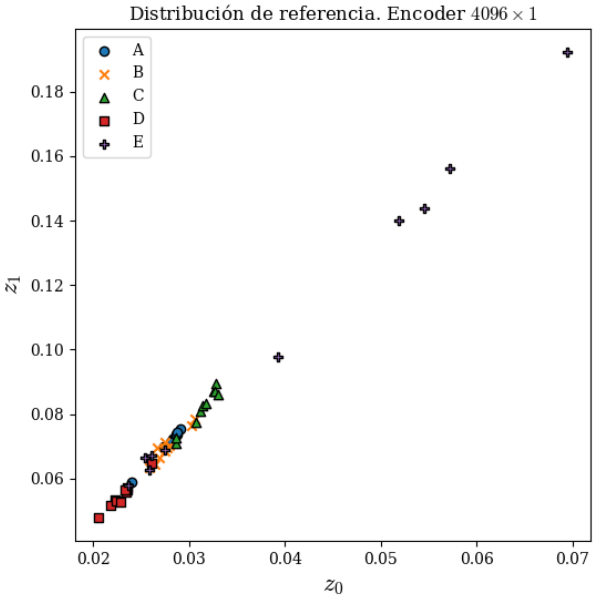

Figura 7: Distribución de las señales de los conjuntos A, B, C, D y E para el Encoder 4096×1.

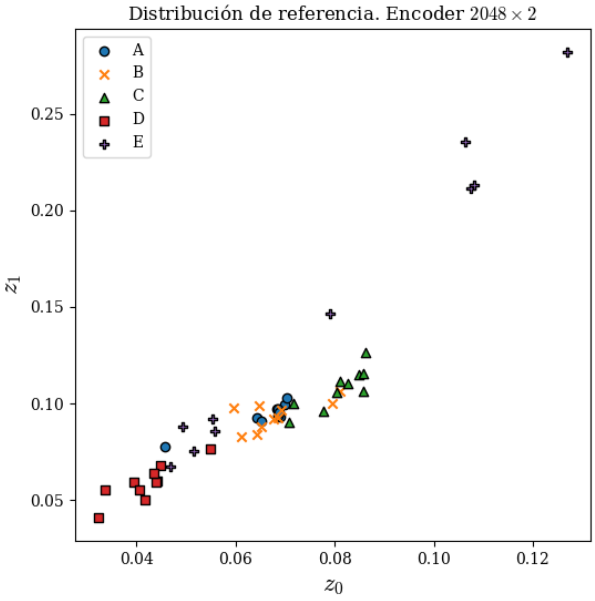

Figura 8: Distribución de las señales de los conjuntos A, B, C, D y E para el Encoder 2048×2.

están ubicados en la misma región relativa del espacio $z_0 - z_1$.

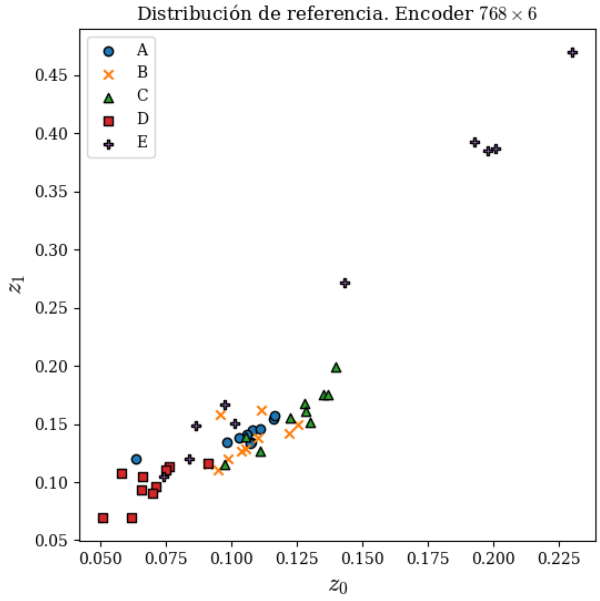

Figura 9: Distribución de las señales de los conjuntos A, B, C, D y E para el Encoder 768×6.

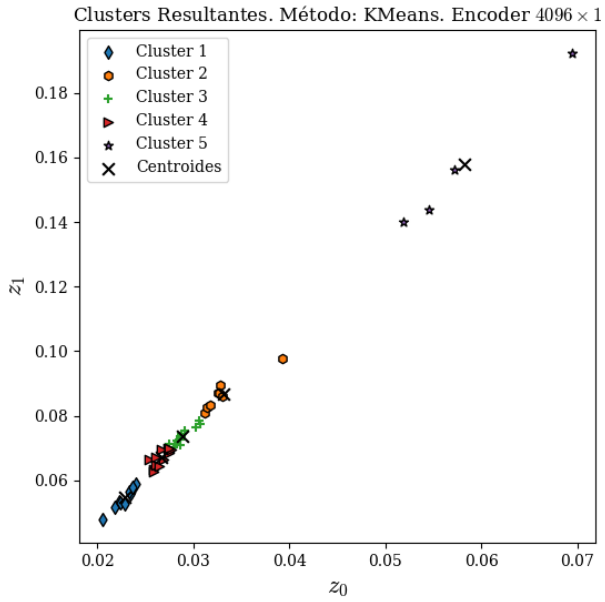

Figura 10: Grupos obtenidos con el método *K*-means para el Encoder 4096×1.

3.3.2. *Agrupamiento con el método SVC*

Los ajustes para el entrenamiento de las tres clasificadores de una sola clase y del etiquetamiento basado en conos se muestran

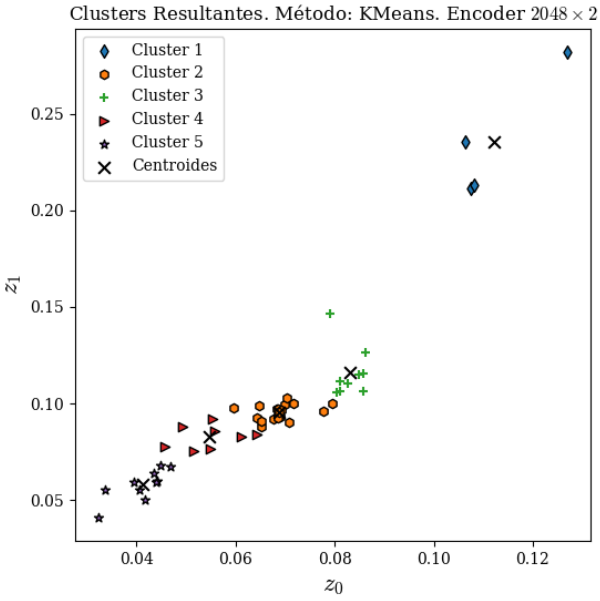

Figura 11: Grupos obtenidos con el método *K*-means para el Encoder 2048×2.

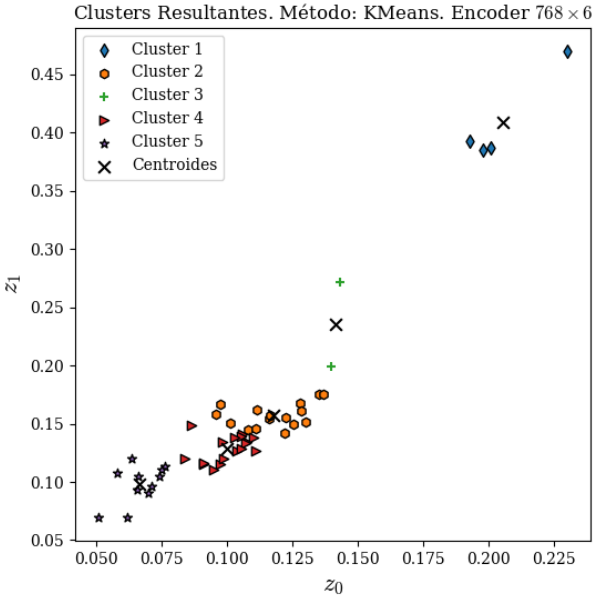

Figura 12: Grupos obtenidos con el método *K*-means para el Encoder 768×6.

en la Tabla 3. Los valores se ajustaron para tener un máximo de cinco grupos. El factor de etiquetamiento lo que hace es modificar el rango de alcance (distancia) de influencia de los clusters definidos por los vectores de soporte. La Tabla 3 también muestra el número de vectores de soporte y el número de grupos obtenidos con el método SVC para cada una de las representaciones latentes de los autoencoders. Las Figuras 13, 14 y 15 muestran los clusters definidos para cada autoencoder.

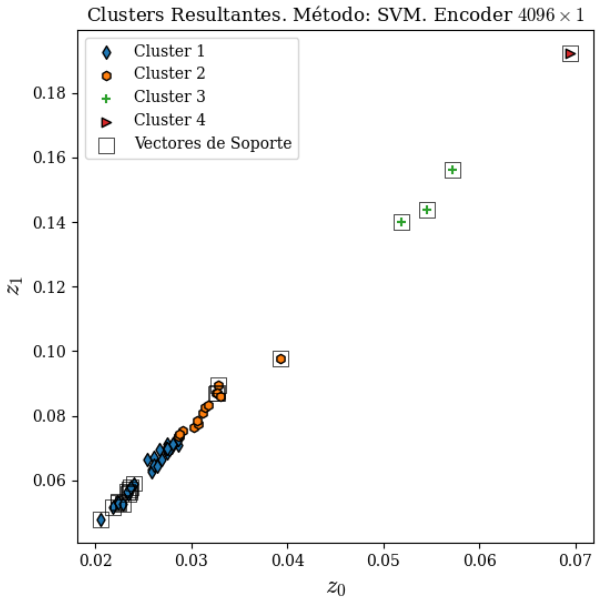

Figura 13: Grupos obtenidos con el método SVC para el Encoder 4096×1.

La Tabla 4 muestra el desempeño de los métodos de agrupamiento basados en *K*-means y SVC, aplicando la formulación de la exactitud basada en el algoritmo húngaro (*ExacHung*) y la información mutua normalizada (*IMN*). La medida *ExacHung* no está disponible para el método SVC, para los autoencoders 4096×1 y 2048×2, porque el número de grupos hallados fue 4.

Tabla 4: Desempeño de los algoritmos *K*-means y SVC

Encoder	Medidas de desempeño			
	ExacHung		*IMN*	
	K-means	SVC	*K*-means	SVC
4096×1	0,66	-	0,52	0,28
2048×2	0,62	-	0,54	0,42
768×6	0,50	0,44	0,36	0,36

Se puede observar en la Tabla 4 que el mejor agrupamiento lo obtuvo el método *K*-means con

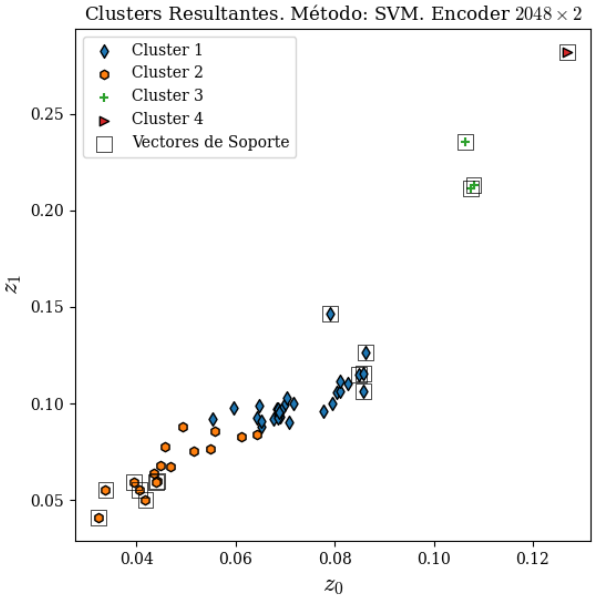

Figura 14: Grupos obtenidos con el método SVC para el Encoder 2048×2.

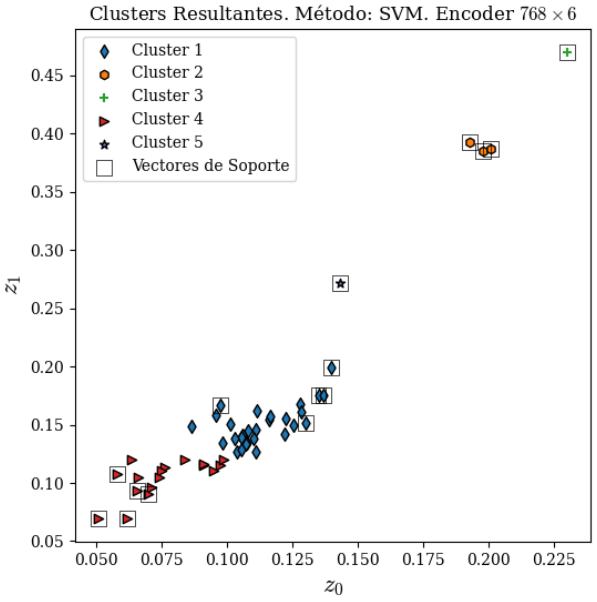

Figura 15: Grupos obtenidos con el método SVC para el Encoder 768×6.

las entradas latentes del autoencoder 4096×1, lo que confirma que la mejor representación latente fue aquella con el menor error de reconstrucción (0,22 %). No obstante, el agrupamiento obtenido con el autoencoder 2048×2 también tuvo excelente desempeño.

4. Análisis y discusión

Se puede verificar, de los resultados obtenidos, que el error de reconstrucción de los autoencoders disminuye cuando se entrena con la señal completa, en comparación a cuando se entrena con la señal segmentada, esto puede estar influenciado por el proceso de normalización de las señales, ya que cada columna se normaliza como si fuera una señal independiente (cierto solo para las señales sin segmentar), lo que no es cierto para las señales segmentadas. A pesar de lo anterior, el error del peor caso (señal segmentada en 6 segmentos) es 1,47 % evaluado sobre el conjunto de pruebas, lo que indica una buena capacidad de reconstrucción de las señales por los tres autoencoders.

Para poder tener una representación geométrica visualizable de las variables latentes (salida del autoencoder) se hallaron las medias (z_0) y las desviaciones estándar (z_1) de dichas variables latentes, para obtener un vector bidimensional y poder visualizar su distribución sobre el plano $z_0 - z_1$. Se observa, casi una invariabilidad en la posición relativa de los puntos o señales pertenecientes a los conjuntos A, B, C, D y E, lo que puede deberse a la buena representatividad de la variables latentes a las señales originales, obtenidas por cada autoencoder. El efecto observable es una disminución de la media (eje z_0) y la desviación estándar (eje z_1) con el aumento de la longitud de la señal y/o la disminución de su dimensión, lo cual se comprueba por simple inspección al observar los rangos de los ejes z_0 y z_1 de las Figuras 7, 8 y 9.

El proceso de agrupamiento obtenido por cada método (K-means y SVC) mostraron resultados con buen desempeño, lo que se confirma visualmente. Pero de acuerdo a las medidas de desempeño, los mejores resultados de agrupamiento se lograron con el método K-means y los autoencoders 4096×1 y 2048×2. Se puede ver que el efecto de la longitud del segmento de señal influencia la detección de los grupos, ya que el aumento o disminución de la distancia relativa entre los puntos en el plano $z_0 - z_1$, facilita o dificulta el proceso de agrupamiento. El método SVC pudo determinar 5 grupos solo para

el caso de la señal de longitud de 768 muestras, y solo 4 grupos para las señales de longitudes de 4096 y 2048 muestras, pero con grupos de solo de 1 miembro, correspondiente a los puntos muy alejados del resto.

5. Conclusiones

Este trabajo presentó la propuesta de un esquema de agrupamiento de señales EEG basados en variables latentes obtenidas de un autoencoder convolucional. Se pudo verificar que, independientemente de la longitud de la señal de entrada al autoencoder, el error de reconstrucción obtenido fue bastante bajo, menos del 1,5 %, aún para el caso más exigente (señal de 768×6).

Se verificó la fortaleza de los autoencoders como extractor de rasgos no supervisado, al obtener una buena representación latente de las señales de entrada.

Los algoritmos de agrupamientos obtuvieron unos grupos con miembros (puntos) consistentemente homogéneos, y con buenas medidas de desempeño para el método *K*-means. El mejor desempeño de agrupamiento se obtuvo con el método *K*-means con la representación latente de las señales obtenidas con el autoencoder 4096×1.

6. Referencias

[1] A. Neligan and J. W. Sander, *Epidemiology of Seizures and Epilepsy*. John Wiley & Sons, Ltd, 2014, ch. 4, pp. 28–32.

[2] D. Shorvon, Simon, *Handbook of epilepsy treatment. Forms, causes and therapy in children and adults*, 2nd ed. Blackwell Publishing, 2005.

[3] S. Miyamoto, H. Ichihashi, and K. Honda, *Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications*, 1st ed. Springer Publishing Company, 2008.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39 501–39 514, August 2018.

[6] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[7] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125–137, march 2001.

[8] S.-H. Lee and K. M. Daniels, "Gaussian kernel widths selection and fast cluster labeling for support vector clustering," University of Massachusetts Lowell, Tech. Rep., 2005.

[9] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[10] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt *et al.*, "Support vector method for novelty detection." in *NIPS*, vol. 12. MIT Press, 1999, pp. 582–588.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer, 2000, vol. 1.

[12] L. Rokach, "A survey of clustering algorithms," in *Data Mining and Knowledge Discovery Handbook*, 2nd ed., O. Maimon and L. Rokach, Eds. Springer, 2016, ch. 14, pp. 269–297.

[13] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955.

[14] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, February 2009.

[15] R. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 061907, pp. 061 907–1–061 907–8, 2001.