Universidad de
SanAndrés

I302 - Aprendizaje Automático y Aprendizaje Profundo

Trabajo Práctico Final: Detección de anomalías en señales ECG

Ana Paula Tissera y Naomi Couriel

2 de julio de 2025

Ingeniería en Inteligencia Artificial

Resumen

En este trabajo abordamos el problema de detección automática de anomalías en señales de ECG, con el objetivo de asistir en el diagnóstico clínico. Para ello, desarrollamos un pipeline híbrido compuesto por un Autoencoder Variacional 1D (VAE) y un clasificador XGBoost, alimentado por la latente del VAE y su error de reconstrucción. El modelo se entrena exclusivamente con registros normales de las bases PTB-XL y Chapman-Shaoxing, utilizando segmentos de la derivación II y coeficientes morfológicos extraídos mediante un modelo FMM. Evaluamos el sistema con métricas estándar (AUC, Precision, Recall, F1, Accuracy) en un conjunto de prueba no visto, obteniendo un AUC de 0.9151, Precision de 0.92 y una F1 de 0.80. Los resultados validan el enfoque y motivan mejoras como atención o aumento de datos.

1. Introducción

Las enfermedades cardiovasculares son una de las principales causas de mortalidad a nivel mundial, y el electrocardiograma (ECG) es la herramienta no invasiva más utilizada para su diagnóstico. Sin embargo, su análisis manual es lento, costoso y dependiente de especialistas, dificultando su aplicación a gran escala. Este trabajo aborda la detección de anomalías en señales de ECG, buscando filtrar automáticamente los registros normales de aquellos que requieren revisión clínica.

Proponemos un pipeline híbrido en dos etapas que combina aprendizaje no supervisado y supervisado, utilizando registros de las bases públicas PTB-XL y Chapman-Shaoxing. La entrada al sistema son segmentos de 2048 muestras de derivación II, junto con un vector de 21 coeficientes morfológicos (FMM). La salida es una clasificación binaria: *Normal* o *Anómalo*.

Primero, entrenamos un Autoencoder Variacional 1D (VAE) convolucional con segmentos normales, utilizando la derivación II y ventanas de 2048 muestras, complementadas con coeficientes FMM que caracterizan la morfología de ondas P, Q, R, S y T. En este contexto, las anomalías generan errores de reconstrucción elevados con respecto a las señales sanas.

Luego, combinamos los vectores latentes del encoder con el error de reconstrucción para entrenar un clasificador XGBoost que afina la decisión binaria. La arquitectura VAE consiste en un encoder convolucional con bloques de reducción temporal y un decoder simétrico. La pérdida incluye el MSE de reconstrucción y la divergencia KL, regularizando el espacio latente.

Este enfoque aprovecha la capacidad del VAE para modelar la variabilidad típica de ECGs normales, sumando la precisión de XGBoost, y ofrece una solución robusta y escalable para la automatización del análisis de ECG.

2. Conjunto de datos y características

Para el desarrollo de este trabajo se utilizaron dos bases de datos de ECG públicas y open source: PTB-XL y Chapman-Shaoxing. La combinación de ambas nos permitió contar con un conjunto de datos diverso y de gran tamaño para entrenar y validar nuestro modelo.

2.1. Fuentes de Datos

El conjunto de datos PTB-XL contiene 21,837 registros de ECG de 12 derivaciones (10s a 500 Hz) de 18,885 pacientes. Seguimos las divisiones de datos recomendadas por los autores del dataset [3]. El dataset Chapman-Shaoxing aporta 10,646 registros adicionales con características similares [4]. Para este, aplicamos una división inicial del 70 % para entrenamiento y 30 % para pruebas, extrayendo después un subconjunto de validación del de entrenamiento.

2.2. Preprocesamiento y Extrcción de Características

El preprocesamiento de las señales y la extracción de coeficientes morfológicos en este trabajo se basan en la metodología desarrollada en el repositorio FMM-Head [1], que provee conjuntos de datos ya segmentados, filtrados y anotados con los parámetros del modelo FMM. Específicamente, utilizamos las versiones `ptb_xl_fmm` y `shaoxing_fmm`, que contienen la señal de ECG ya centrada en picos R, recortada a 2048 muestras y acompañada de un vector de 21 coeficientes por latido.

Siguiendo el enfoque del repositorio original, se seleccionan las señales de la derivación II por coincidir su eje de registro ($\sim$+60°) con la dirección natural de la despolarización auricular y ventricular, facilitando la identificación clara de la onda P (despolarización auricular), el complejo QRS (despolarización ventricular) y la onda T (repolarización ventricular), cuyos cambios en forma o duración, como anomalías en la anchura del QRS o inversión de la T, pueden indicar isquemia o bloqueos de conducción, entre otros. A continuación, dichas señales se someten a un filtro pasa-bajo para eliminar la deriva de línea base causada por respiración o movimiento.

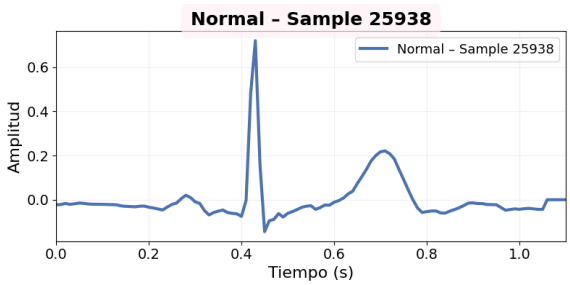
Luego, se detectan los picos R mediante las ecuaciones de Pan–Tompkins y se define el intervalo RR como la diferencia de tiempo entre picos R sucesivos. Cada latido se segmenta centrado en el pico R, tomando el 40 % del intervalo RR anterior y el 60 % del siguiente, y se rellena con ceros hasta alcanzar una longitud fija de 2048 muestras (múltiplo de *batch-size*) para alimentar el modelo.

La forma de cada latido se describe mediante el modelo de Frecuencia Modulada de Möbius (FMM), que aproxima la señal como la suma de cinco ondas moduladas (P, Q, R, S y T):

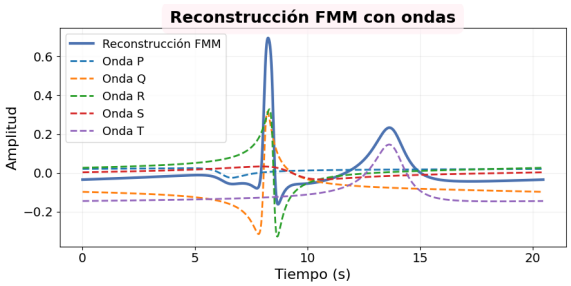
$$x(t) \;=\; \sum_{j=1}^{5} A_j \,\sin\!\Big(\omega_j t \;+\; \beta_j \,\arctan\!\big(\tan\!\big(\tfrac{t-\alpha_j}{2}\big)\big)\Big) \;+\; C,$$

donde A_j es la amplitud, ω_j la frecuencia, α_j la fase central, β_j el parámetro de asimetría y C un offset global. De este modo, cada onda aporta cinco parámetros y el offset suma uno, dando un total de 21 coeficientes morfológicos. Para evitar discontinuidades angulares, cada fase α_j se codifica como $\big(\cos\alpha_j,\ \sin\alpha_j\big)$, y todos los parámetros se ordenan según la onda correspondiente, de P a T [1].

La señal modulada a partir de esta suma captura la forma esencial de cada latido, preservando las características relevantes para la detección de anomalías —como se evidencia en las Figuras 1 y 2— y, al mismo tiempo, aporta una representación más informativa al modelo.

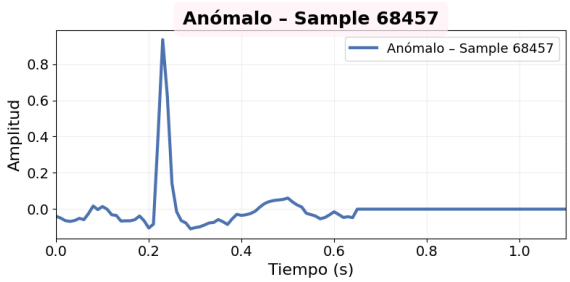


(a) Señal sana

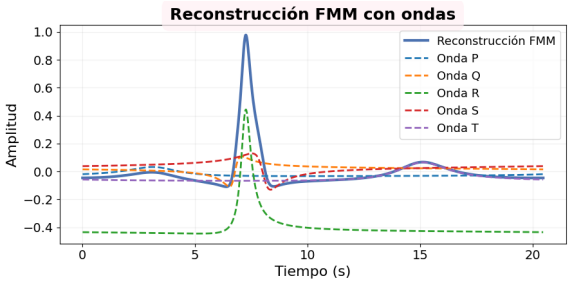


(b) FMM sano

Figura 1: Comparación entre señal sana y su reconstrucción FMM.



(a) Señal anómala



(b) FMM anómalo

Figura 2: Comparación entre señal anómala y su reconstrucción FMM.

Además, este vector de 21 características se utiliza como una entrada condicional para el VAE junto con la señal del ECG modulada. De este modo, el VAE no solo aprende a reconstruir en base a la señal, sino que lo hace también basándose en las características morfológicas explícitas, mejorando su capacidad de representación.

Finalmente, para evitar filtraciones de información entre conjuntos, calculamos la media y la desviación estándar exclusivamente sobre las ventanas normales del conjunto de entrenamiento. Estos valores se utilizaron para estandarizar todas las ventanas (entrenamiento, validación y prueba) a media cero y varianza unitaria, garantizando así la integridad del proceso de validación.

3. Metodología

En este apartado describimos con detalle los algoritmos que componen nuestro pipeline híbrido: el Autoencoder Variacional 1D (VAE) convolucional y el clasificador supervisado XGBoost.

3.1. Autoencoder Variacional 1D Condicional

El VAE se define como un modelo generativo probabilístico con un *encoder* $q_\phi(\mathbf{z} \mid \mathbf{x}_s, \mathbf{x}_f)$ y un *decoder* $p_\theta(\mathbf{x}_s \mid \mathbf{z})$. El *encoder* procesa la señal ECG $\mathbf{x}_s$ y el vector de coeficientes FMM $\mathbf{x}_f$ para mapearlos a una distribución latente. Su arquitectura es la siguiente:

- Una capa inicial Conv1D$(1 \to 32,\ k = 19,\ s = 2,\ p = 9)$ con activación LeakyReLU.
- $n_{\text{blocks}} = 3$ bloques residuales 1D ResBlock1D$(32,\ k = 3,\ \text{dilation} = 1, 2, 4)$ que usan dilataciones crecientes para ampliar el campo receptivo sin alterar la resolución temporal.

- Un aplanado (*flatten*) de la salida convolucional, que se concatena con el vector de coeficientes FMM.
- Dos capas densas que proyectan el vector concatenado para obtener los parámetros de la distribución latente: la media $\boldsymbol{\mu}$ y el logaritmo de la varianza $\log \boldsymbol{\sigma}^2$, ambas de dimensión $d_z = 64$.

Mediante el truco de reparametrización se muestrea el vector latente $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, con $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. El *decoder* reconstruye la señal $\hat{\mathbf{x}}_s$ a partir de $\mathbf{z}$ con arquitectura simétrica.

La función de pérdida minimizada es la *Evidence Lower Bound* (ELBO):

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\|\mathbf{x}_s - \hat{\mathbf{x}}_s\|_2^2] + \beta \cdot D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \parallel \mathcal{N}(0, I)\big) \tag{1}$$

donde el primer término es el error de reconstrucción (MSE) y el segundo la divergencia de Kullback-Leibler, ponderada por el hiperparámetro β.

3.2. Clasificador XGBoost

Una vez entrenado el VAE, se extrae un vector de características $\mathbf{c} = [\boldsymbol{\mu}, e] \in \mathbb{R}^{65}$ para cada señal, donde $\boldsymbol{\mu}$ es el vector latente y e el error de reconstrucción. Este vector se utiliza para entrenar un clasificador XGBoost que realiza la predicción final (*Normal* vs. *Anómalo*).

Para convertir las probabilidades continuas de XGBoost en etiquetas binarias, se calculó la curva ROC (3.3) sobre el conjunto de desarrollo y se seleccionó el umbral correspondiente al punto de Youden (máxima sensibilidad más especificidad menos uno). Este umbral optimiza el balance entre detectar anomalías verdaderas y minimizar falsos positivos.

3.3. Métricas de Evaluación

El rendimiento del clasificador se evaluó con las siguientes métricas estándar para clasificación binaria:

- **Matriz de Confusión**: permite visualizar la distribución de los casos en verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN), proporcionando una representación directa del comportamiento del modelo.

	Predicho: Positivo	**Predicho: Negativo**
Real: Positivo	TP	FN
Real: Negativo	FP	TN

- **Accuracy**: mide la proporción de predicciones correctas sobre el total de muestras, definida como

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

- **Precision**: capacidad del modelo para no etiquetar una muestra normal como anómala. Se calcula como

$$P = \frac{TP}{TP+FP} \tag{3}$$

- **Recall (Sensibilidad)**: capacidad del modelo para encontrar todas las muestras anómalas. Se calcula como

$$R = \frac{TP}{TP+FN} \tag{4}$$

- **Curva Precision–Recall**: muestra en el eje horizontal el recall y en el vertical la precision al variar el umbral de decisión; sirve para visualizar el trade-off entre ambas métricas y elegir el umbral óptimo.

- **F1 Score**: combina precisión y recall en una única métrica balanceada.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{5}$$

- **Curva ROC**: gráfico de la TPR (True Positive Rate) frente a la FPR (False Positive Rate) al variar el umbral de decisión; permite visualizar el compromiso entre detectar positivos reales y evitar falsos positivos.

- **AUC-ROC**: es el área bajo la curva ROC, obtenida por integración trapezoidal sobre la relación entre TPR y FPR; mide la capacidad de distinguir entre clases a distintos umbrales.

4. Experimentos, Resultados y Discusión

En esta sección presentamos la configuración experimental, la selección de hiperparámetros reales, las métricas empleadas y los resultados tanto en validación como en el conjunto de prueba final, junto con un análisis de errores.

4.1. Configuración experimental y selección de hiperparámetros

Para encontrar la configuración óptima del VAE, se realizó una búsqueda de hiperparámetros (*hyperparameter sweep*), entrenando cada combinación durante 15 épocas. El criterio fue maximizar el AUC del clasificador XGBoost resultante en el conjunto de validación. Los rangos explorados fueron:

- Regularización $\beta \in \{1{,}0, 4{,}0, 10{,}0\}$.
- Tasa de aprendizaje $\ell_r \in \{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$.
- Dimensión latente $d_z \in \{16, 32, 64\}$.
- Número de bloques convolucionales $n_{\text{blocks}} \in \{2, 3\}$.

Se utilizó el optimizador Adam con un *batch size* de 64. La mejor configuración, que alcanzó un AUC de XGBoost de 0.9331 en validación, fue $\beta = 1{,}0$, $\ell_r = 5 \times 10^{-4}$, $d_z = 64$ y $n_{\text{blocks}} = 3$. El modelo final se re-entrenó durante 35 épocas con estos parámetros.

4.2. Exploración de los Datos y Características FMM

El pipeline de este trabajo incorpora como conocimiento previo (*prior knowledge*) un conjunto de 21 características morfológicas extraídas mediante un Modelo Basado en Fourier (FMM, por sus siglas en inglés). Este modelo, propuesto por Verardo et al. (2023), analiza la morfología de cada latido para caracterizar las ondas P, Q, R, S y T como se explicó en la sección 2.2. Estos coeficientes se introducen en el *encoder* del VAE para condicionar el aprendizaje de la representación latente.

4.3. Análisis de Convergencia del Entrenamiento

Para validar la estabilidad del componente de aprendizaje no supervisado, se analizó la evolución de la función de pérdida del VAE durante su re-entrenamiento final de 35 épocas. La Figura 3 muestra una disminución suave y sostenida de la pérdida total (ELBO), así como de sus componentes de reconstrucción (MSE) y regularización (KL). Esto indica una convergencia estable, sin signos de inestabilidad o divergencia, lo que da confianza en la calidad de las características extraídas por el VAE.

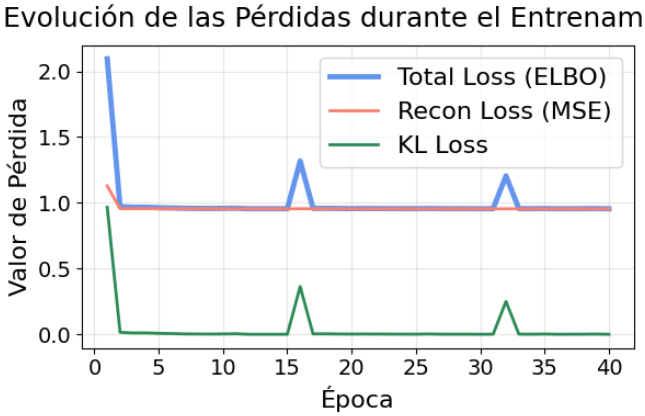


Figura 3: Curvas de pérdida del VAE durante las 35 épocas de entrenamiento final. Se observa una convergencia relativamente estable.

4.4. Resultados en validación y test final

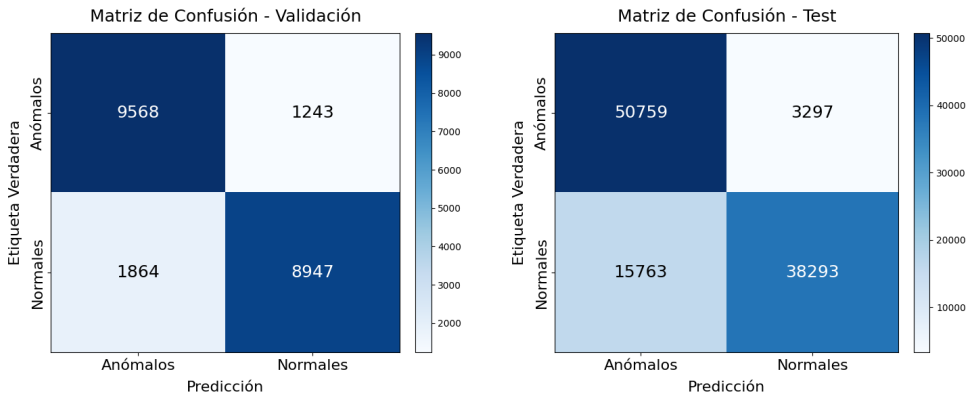
Los resultados numéricos del modelo final se resumen en la Tabla 1. Se observa un rendimiento general alto, con un AUC superior a 0.91 en el conjunto de prueba.

Cuadro 1: Métricas en validación y test final

Métrica	Validación	Test final
AUC-ROC	0.9235	0.9151
Precision	0.8780	0.9207
Recall	0.8276	0.7084
F1-score	0.8521	0.8007
Accuracy	0.8563	0.8237

Para evitar el sesgo hacia la clase mayoritaria, el conjunto de prueba fue balanceado mediante *undersampling*, equiparando la cantidad de segmentos normales y anómalos.

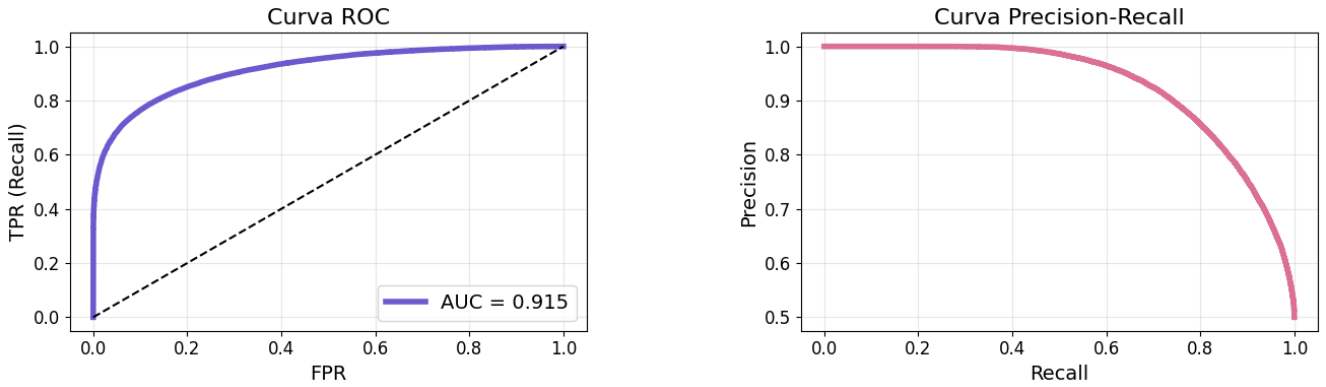
Las matrices de confusión (Figura 4) permiten visualizar la distribución de los errores. En el conjunto de test, el modelo identificó correctamente 50759 anomalías, pero falló en detectar 15763 normales, lo que explica el valor de Recall. Por otro lado, solo 3297 señales anómalas fueron incorrectamente clasificadas como normales, validando la alta Precisión del modelo.



(a) Validación (b) Test final

Figura 4: Matrices de confusión en validación y test final.

En el conjunto de desarrollo, el punto de Youden sobre la curva ROC dio un umbral óptimo de 0,584 (TPR = 0.839, FPR = 0.061). Este mismo umbral se utilizó para convertir las probabilidades continuas en etiquetas binarias a la hora de computar las matrices de confusión y las métricas finales en validación y test.



(a) Curva ROC (Test) (b) Curva Precision-Recall (Test)

Figura 5: Curvas de rendimiento en el conjunto de prueba. El área bajo la curva ROC (izquierda) confirma la buena capacidad de clasificación del modelo.

Las curvas ROC y Precision-Recall (PR) del conjunto de test (Figura 5) muestran que el modelo mantiene un buen poder discriminativo, con un AUC de 0.9151 muy por encima de un clasificador aleatorio.

4.5. Discusión

Análisis cuantitativo

El pipeline híbrido VAE+XGBoost demuestra un rendimiento sólido, superando ampliamente el umbral basado únicamente en reconstrucción. El modelo alcanzó un AUC superior a 0.91 y una Precisión por encima de 0.92 en el conjunto de prueba. La reducción de Recall en test (0.7084) indica que ciertos casos de anomalía más sutiles son menos detectados. La mayoría de los falsos positivos provienen de artefactos de ruido, lo que sugiere que el sistema prioriza evitar falsos positivos aunque a costa de perder sensibilidad.

5. Conclusión y trabajo futuro

Este comportamiento refleja un trade-off entre Precisión y Recall: el modelo se comporta de manera conservadora. Podrían explorarse técnicas como mecanismos de atención o augmentación sintética para mejorar la capacidad de detección de anomalías sutiles. Asimismo, un preprocesamiento más agresivo podría reducir el impacto de señales ruidosas. En conjunto, el enfoque propuesto ofrece una base sólida, y abre el camino hacia mejoras de robustez frente a variaciones de señal y ruido, demostrando ser una herramienta prometedora para asistir el análisis automático de ECGs en contextos clínicos reales.

Referencias

[1] Giacomo Verardo, Magnus Boman, Samuel Bruchfeld, Marco Chiesa, Sabine Koch, Gerald Q. Maguire Jr., y Dejan Kostic. *FMM-Head: Enhancing Autoencoder-based ECG anomaly detection with prior knowledge*, 2023.

[2] Jong-Hwan Jang, Tae Young Kim, Hong-Seok Lim, y Dukyong Yoon. *Unsupervised feature learning for electrocardiogram data using the convolutional variational autoencoder*. PLOS ONE, vol. 16, no. 12, p. e0260612, 2021. `https://doi.org/10.1371/journal.pone.0260612`

[3] Patrick Wagner, Nico Strodthoff, Robert Bousseljot, Daniel Kreiseler, Frank Lunze, Wojciech Samek, y Thomas Schaeffter. *PTB-XL: A Large Publicly Available ECG Dataset*. Scientific Data, 7, 154 (2020). `https://doi.org/10.1038/s41597-020-0495-6`

[4] Jie Zheng, Jinbo Zhang, Sibao Danioko, et al. *A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients*. Scientific Data, 7, 48 (2020). `https://doi.org/10.1038/s41597-020-0381-4`

[5] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. NeurIPS.

[6] Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[8] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). *Array programming with NumPy*. Nature, 585(7825), 357–362.

[9] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference.

[10] Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3), 90–95.

Fuentes de datos

1. **PTB-XL ECG Dataset, versión 1.0.3.** Patrick Wagner et al. (2020). *PTB-XL: A Large Publicly Available ECG Dataset. Scientific Data.* Disponible en PhysioNet: `https://doi.org/10.13026/kfzx-aw45`. Citado como [3].

2. **Chapman-Shaoxing Database de ECG 12 derivados.** Jie Zheng et al. (2020). *A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Scientific Data.* Citado como [4].

3. **Conjuntos preprocesados FMM (Fast Multi-resolution Matching).** Archivos disponibles públicamente:

   - `https://drive.google.com/uc?id=1nYRvbVYJJXPbCwKEqOIeCLnMXdIDAka7`
   - `https://drive.google.com/uc?id=1FjvmVb8-PnpDdoBwv-Eqb89tYFOCx9KW`

   (Estos archivos contienen las señales originales transformadas mediante FMM)