

## CLASIFICACIÓN DE SEÑALES DE GLUCEMIA PARA LA DETECCIÓN DE DIABETES MELLITUS TIPO 2 MEDIANTE EL USO DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO

### CLASSIFICATION OF BLOOD GLUCOSE SIGNALS FOR THE DETECTION OF TYPE 2 DIABETES MELLITUS USING A GEOMETRIC FEATURE SPACE AND MACHINE LEARNING ALGORITHMS

M. García Blesa<sup>a</sup>, J. Vorobioff<sup>b</sup> y W. Legnani<sup>a</sup>

<sup>a</sup>Centro de Procesamiento de Señales e Imágenes, Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Medrano 951 (C1179AAQ) CABA, Argentina, [hgarciablesa@frba.utn.edu.ar](mailto:hgarciablesa@frba.utn.edu.ar),  
<https://www.frba.utn.edu.ar/>

<sup>b</sup>Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, <https://www.frba.utn.edu.ar/>

**Palabras clave:** amplitud, ángulo de zenit, factor de forma, diferenciación de señales

**Resumen.** Este estudio propone un nuevo enfoque para la detección temprana de diabetes mellitus tipo 2 mediante el análisis automatizado de señales fisiológicas. El método desarrollado se basa en la construcción de un espacio de características geométricas generado a partir de parámetros extraídos de señales de pacientes, y la aplicación posterior de algoritmos de aprendizaje automático sobre dicho espacio. Los resultados experimentales demuestran una elevada eficacia diagnóstica, con valores de F1-score que alcanzan 1.000 en las mejores configuraciones, respaldados por métricas complementarias de sensibilidad (en el rango [0.960–1.000]), especificidad (igual o superior a 0.984) y precisión balanceada (igual o mayor a 0.983). Estos resultados confirman que la técnica propuesta ofrece un sistema novedoso a la vez que computacionalmente eficiente para la identificación de patrones asociados a la diabetes mellitus tipo 2.

**Keywords:** amplitud, ángulo de zenit, factor de forma, diferenciación de señales

**Abstract.** This study proposes a new approach for the early detection of type 2 diabetes mellitus through the automated analysis of physiological signals. The method developed is based on the construction of a geometric feature space generated from parameters extracted from patient signals, and the subsequent application of machine learning algorithms on this space. Experimental results demonstrate high diagnostic efficiency, with F1-score values reaching 1.000 in the best configurations, supported by complementary metrics of sensitivity (ranging [0.960–1.000]), specificity (equal to or higher than 0.984) and balanced accuracy (equal to or higher than 0.983). These results confirm that the proposed technique offers a robust and reliable system for the identification of patterns associated with type 2 diabetes mellitus.

## 1. INTRODUCCIÓN

La clasificación de la diabetes ha pasado de incluir únicamente dos tipos (tipo 1 y tipo 2) a reconocer más de 50 subcategorías con distintos orígenes y vinculaciones con otras patologías. En particular la diabetes tipo 2 surge debido a una combinación de resistencia a la insulina y producción insuficiente de esta hormona, condiciones influenciadas por el sobrepeso, la inactividad física y el envejecimiento. La obesidad, que antes se asociaba más a los adultos, es ahora frecuente en personas de todas las edades, lo que contribuye a la superposición de factores entre la diabetes tipo 1 y tipo 2. Por otro lado, la diabetes gestacional ocurre exclusivamente durante el embarazo y puede ser un indicio de un futuro diagnóstico de diabetes tipo 2.

Los estudios en diversas poblaciones han revelado inconsistencias en las pruebas de detección de la enfermedad. Entre quienes cumplen los criterios de la prueba de hemoglobina glicosilada (HbA1c), solo entre el 27 % y el 98 % también cumplen los de la glucosa plasmática, y de los que cumplen estos últimos, apenas entre el 17 % y el 78 % coinciden con los valores de la prueba HbA1c. Estas diferencias se deben a que cada examen analiza distintos aspectos de la hiperglucemia, los cuales pueden variar según el paciente ([Genuth et al. \(2018\)](#)).

A pesar de que los algoritmos de machine learning (ML) han demostrado tener un considerable potencial predictivo, todavía tienen limitaciones en el caso de la diabetes mellitus tipo 2. A pesar de que los modelos de ML presentan una discriminación sobresaliente (AUROC  $\sim 0,88$ ), el desempeño aún no es adecuado para cambiar la percepción del riesgo futuro en los pacientes con seguridad. Los autores [Abhari et al. \(2023\)](#) subrayan que los modelos de ML, a pesar de ser más flexibles, necesitan un aumento significativo en su capacidad de generalización y precisión; por otro lado, los métodos tradicionales tienen una validez externa baja y dependen de un número reducido de variables. Se destaca la relevancia de crear algoritmos más sólidos y eficaces —por ejemplo, utilizando estructuras avanzadas como redes recurrentes o transformadores— con el fin de mejorar la detección temprana y la categorización de la diabetes tipo 2 en diversas situaciones poblacionales.

Los autores [Kodama et al. \(2022\)](#) indican que, a pesar de que los algoritmos de aprendizaje profundo, las redes neuronales, los algoritmos como Random Forest y las máquinas de soporte vectorial (SVM) son más eficaces que los métodos estadísticos clásicos, su desempeño está muy vinculado con la representatividad y la calidad de los datos utilizados en el entrenamiento. Asimismo, numerosas estrategias actuales se fundamentan en grupos limitados de variables que no representan el carácter dinámico y no lineal de esta patología. Por lo tanto, los escritores subrayan la importancia de crear métodos de clasificación más sólidos y comprensibles, que sean aptos para incluir señales complejas (como las series de glucemia en el tiempo) y que contribuyan a perfeccionar la identificación temprana de la enfermedad y su caracterización precisa.

La detección temprana de la diabetes mellitus tipo 2 es fundamental por su carácter silencioso en etapas iniciales y su potencial para desarrollar complicaciones graves como neuropatías, nefropatías, retinopatías y enfermedades cardiovasculares cuando no se controla. Al diagnosticarse a tiempo mediante pruebas sencillas (glucemia en ayunas o HbA1c), especialmente en poblaciones de riesgo (mayores de 45 años, personas con obesidad o antecedentes familiares), se puede intervenir con modificaciones en el estilo de vida -dieta equilibrada, actividad física regular y control de peso- que en muchos casos logran revertir la condición antes de requerir tratamiento farmacológico. Esta prevención no solo mejora significativamente la calidad de vida del paciente, sino que reduce sustancialmente los costos asociados al manejo de complicaciones crónicas, convirtiendo a la detección precoz en una estrategia clave de salud pública ([American](#)

Diabetes Association (2023), Herman et al. (2015).)

### 1.1. Método

Mediante el algoritmo propuesto en García Blesa et al. (2025), se procesa la base de datos de señales discretas para obtener el GFS, siguiendo un esquema general sencillo. Como se observa en la figura 1, este espacio se emplea luego como *data-set* en algoritmos de machine learning. A continuación se da una breve flujo grama del trabajo desarrollado. El primer elemento rele-

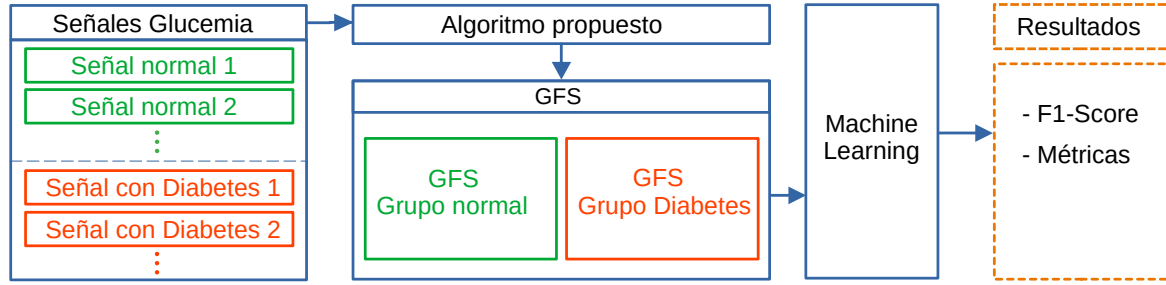


Figura 1: Esquema general del método.

vante para definir el GFS es la amplitud de la señal  $S$ . Esta se calcula considerando los valores extremos  $\min(S)$  y  $\max(S)$  de la siguiente manera:

$$A(S) = \max(S) - \min(S). \quad (1)$$

El segundo elemento, denominado ángulo Zenith ( $\varphi$ ), toma como parámetro el promedio de los valores de  $S$ . Éste se calcula del siguiente modo:

$$\varphi = \arctan\left(\frac{1-p}{0,5}\right) = \arctan(2-2p), \quad (2)$$

donde  $p$  es el promedio de los valores de  $S$ .

Como tercer parámetro, el Factor de Forma cuantifica la distribución de picos y valles en  $S$ . Considerando una señal discreta  $S$  de longitud  $n$ , el procedimiento consiste en: 1) ordenar de menor a mayor los valores de  $S$ , y 2) construir el vector  $O$  con los índices resultantes. Luego si  $\mathcal{P}_n = \{q_1, q_2, \dots, q_n\}$  es el conjunto de los primeros  $n$  números primos, el factor de forma se define como sigue:

$$\mathcal{F}(S) = \prod_{i=1}^n q_i^{o_i}. \quad (3)$$

donde  $\mathcal{F}(S) \in \mathbb{N}$ . Mediante el Teorema Fundamental de la Aritmética, se puede descomponer este parámetro para obtener la configuración de valles y picos de  $S$ .

Dada  $S$  de longitud  $n$ , y dada una dimensión de *embedding*  $d < n$ , se definen los siguientes segmentos  $s$  de longitud  $d$  derivados de  $S$ :

$$s_i = \{a_i, a_{i+1}, \dots, a_{i+d-1}\}, \quad (4)$$

donde  $s_i \subset S$ , e  $i = 1, 2, \dots, n-d+1$ . Cabe destacar que los segmentos  $s_i$  son señales discretas, permitiendo así la implementación de los métodos anteriormente descritos. Cada  $s_i$  (definido

en (4)) queda unívocamente determinado por la tripleta formada por las ecuaciones (1), (2) y (3):

$$s_i \mapsto (A(s_i), \varphi(s_i), \mathcal{F}(s_i)). \quad (5)$$

Simplificando la notación, se escribe  $A_i = A(s_i)$ ,  $\varphi_i = \varphi(s_i)$ , y  $\mathcal{F}_i = \mathcal{F}(s_i)$ . Sustituyendo en la expresión (5), se define el vector de características  $v_i$ :

$$v_i = (A_i, \varphi_i, \mathcal{F}_i). \quad (6)$$

Con el objetivo de obtener un GFS reducido, se discretizan las dos primeras componentes de  $v_i$ . Este mecanismo produce, eventualmente, vectores repetidos. Las dos primeras componentes del vector  $v_i$  –amplitud y ángulo de Zenith– son valores reales, por lo que se considera que la probabilidad de su ocurrencia es muy baja. Para aumentar esta probabilidad, ambos parámetros se discretizan, y los valores calculados se asignan a un rango discreto de tamaño arbitrario. Como criterio de cálculo se busca el menor valor de discretización que logre la mejor clasificación, dado que cuanto menor sea  $t$ , menor será la cantidad de datos y con ello menor tiempo de proceso.

El vector de características con los parámetros amplitud y ángulo de Zenith discretizados se redefine como  $v_i = (A'_i, \varphi'_i, \mathcal{F}_i)$ , donde  $A'_i$  es la amplitud discreta de  $A_i$ , así como  $\varphi'_i$  corresponde al valor discreto de  $\varphi_i$ . El conjunto de todos estos vectores es  $V = \{v_i : i = 1, 2, \dots, n - d + 1\}$ .

Con el propósito de eliminar la información redundante, el conjunto  $V$  se reduce eliminando los elementos repetidos. Así, se considera el conjunto  $V' = \{v'_1, v'_2, \dots, v'_k\}$ , donde  $v_i \in V$  para  $i = 1, 2, \dots, k$ , y para todo  $i, j$  entre 1 y  $k$ , tal que  $i \neq j$ , se cumple que  $v_i \neq v_j$ .

Este conjunto, compuesto por vectores únicos, tiene cardinalidad  $\#V' \leq \#V$ . Es relevante conocer la frecuencia en  $V$  de los vectores de  $V'$ . Para  $i = 1, 2, \dots, k$ , se define la frecuencia  $F_i = \#\{v \in V : v = v_i\}$ . Finalmente se tiene el conjunto  $W$  definido de la siguiente manera:

$$W = \{(v'_i, F_i) : i = 1, 2, \dots, k\}, \quad (7)$$

conteniendo los vectores de  $V'$  y sus respectivas frecuencias en  $V$ . Así, el conjunto  $W$  es el GFS. Si  $\mathcal{S} = \{S_1, S_2, \dots, S_t\}$  es un conjunto de señales, el conjunto  $W$  se asocia con  $\mathcal{S}$  como su GFS.

El posprocesamiento de los GFS's se realizó utilizando los siguientes modelos de ML, ampliamente difundidos en la literatura [Murphy \(2012\)](#). *Support Vector Classifier (SVC)*: Clasificador basado en hiperplanos; configuración: kernel = linear,  $C = 0,025$  (lineal),  $\gamma = 2$ ,  $C = 1$  (no lineal). *Gaussian Process Classifier (GPC)*: Clasificador probabilístico basado en procesos gaussianos; configuración:  $1,0 \times \text{RBF}(1,0)$ . *Decision Tree Classifier (DTC)*: Árbol de decisión; configuración: max\_depth = 5. *Random Forest Classifier (RFC)*: Ensamble de árboles con voto mayoritario; configuración: max\_depth = 5, n\_estimators = 10, max\_features = 1. *Multi-Layer Perceptron Classifier (MLP)*: Red neuronal multicapa; configuración:  $\alpha = 1$ , max\_iter = 1000. *AdaBoost Classifier (AdaBoost)*: Ensamble de clasificadores débiles en un modelo fuerte. *Gaussian Naive Bayes (GNB)*: Clasificador probabilístico que asume independencia condicional entre características. *Logistic Regression (LR)*: Modelo de regresión para clasificación binaria. *K-Neighbors Classifier (KNN)*: Clasificador por voto mayoritario de los  $k = 3$  vecinos más cercanos.

Se calcularon los siguientes parámetros estándar para cada modelo: F1-score, precisión balanceada, sensibilidad, especificidad, valores de PP, NP y tiempo de procesamiento. Adicionalmente se utilizó una validación cruzada 5-Fold para reducir sesgos ([Zihlmann et al. \(2017\)](#)).

Esta configuración de k-fold utiliza el 20 % de los datos del GFS para test y el 80 % para prueba, rotando este procedimiento hasta que todos los folds han sido utilizados como conjunto de prueba. Se aclaró este punto en el documento (James et al. (2021); Géron (2022)). Se calcularon  $F1_{\text{train}}$  y  $F1_{\text{test}}$  para evaluar el posible sobre-ajuste del modelo. Se considera ausencia de sobre-ajuste cuando la diferencia  $F1_{\text{train}} - F1_{\text{test}} < 0,05$ , siguiendo criterios empíricos de brecha de generalización Carrell et al. (2022).

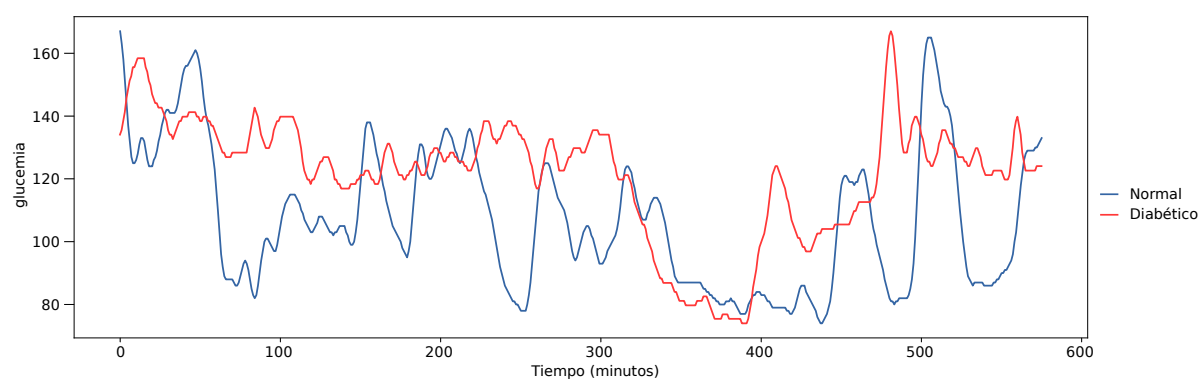


Figura 2: Señales de glucosa. La curva azul corresponde a un caso de test T2DM negativo, y la roja a un caso de test T2DM positivo o de paciente con diabetes Mellitus tipo 2. Los valores corresponden a mediciones tomadas cada 5 minutos durante un total de 24 horas.

La base de datos utilizada se obtuvo de Rodríguez de Castro et al. (2017), la cual incluyó a 208 participantes reclutados en la unidad de hipertensión arterial y riesgo cardiovascular del Hospital Universitario de Móstoles (Madrid) entre enero de 2012 y mayo de 2015. Los sujetos cumplían los siguientes requisitos: tener entre 18 y 85 años, presentar diagnóstico de hipertensión arterial primaria y no tener antecedentes de diabetes mellitus ni recibir medicación. Dado que algunas señales presentaban lagunas o baches de valores de glucosa, se excluyeron del análisis, dando un total de 16 señales con test T2DM positivo de diabetes mellitus tipo 2, y 158 de señales con test T2DM negativo. Las señales fueron etiquetadas como 'NOR' para representar señales no patológicas –referidas aquí como 'señales normales'– y 'DIA' para señales que presentan test T2DM positivo. Los valores corresponden a mediciones tomadas cada 5 minutos durante 24 horas. La figura 2 representa la gráfica de dos de las señales analizadas.

Los parámetros iniciales para nuestro algoritmo fueron definidos arbitrariamente en dos conjuntos: el conjunto  $D$  correspondiente a los valores de la dimensión de *embedding* cuyas cantidades son  $D = \{3, 4, 5, 6\}$ , y el conjunto  $T$  de factores de discretización cuyos valores son  $T = \{10, 20, 30, 50\}$ . Denotamos como  $[d, t]$  al par de parámetros de entrada. Se calcularon un total de 16 casos combinando los elementos de ambos conjuntos:  $[3, 10], \dots, [3, 50], [4, 10], \dots, [4, 50], [5, 10], \dots, [5, 50], [6, 10], \dots, [6, 50]$ , generando un espacio de características respectivo para cada caso. Luego, se aplicaron los algoritmos ML (10 algoritmos en total) a cada espacio de características obteniendo un total de 160 resultados.

## 2. RESULTADOS

De las señales analizadas se obtuvo un total de 16 espacios de características. Se puede observar que, a pesar de que la cantidad de señales NOR y DIA es dispar, la cantidad de vectores característicos encontrada no conserva dichas diferencias. La tabla 1 muestra la cantidad de vectores característicos obtenidos para cada caso  $[d, t]$ .

$d$	$t$	Cant. de vectores NOR	Cant. de vectores DIA
3	10	13	18
3	20	27	23
3	30	28	26
3	50	42	59
4	10	33	49
4	20	66	56
4	30	80	111
4	50	131	128
5	10	84	82
5	20	124	119
5	30	90	93
5	50	186	265
6	10	195	191
6	30	204	304
6	50	281	282

Tabla 1: Cantidad de vectores característicos encontrados en cada GFS.

De los 160 casos analizados, 22 de ellos obtuvo F1-score por encima de 0,9. En dicho grupo se encuentran sólo dos modelos: AdaBoost y Decision Tree. La tabla 2 presenta los 10 mejores resultados según el valor de *F1-score*. Se destaca que la dimensión de *embedding*  $d = 4$  ofrece el mayor rendimiento, como se detalla en la tabla 3. En particular, la configuración *AdaBoost* con  $d = 4$  y  $t = 20$  alcanza valores máximos en todas las métricas evaluadas: *F1-score*, precisión balanceada, sensibilidad y especificidad. Aunque las configuraciones con  $d = 4$  mantienen un *F1-score* superior a 0,989, un incremento a  $d = 6$  conlleva una ligera disminución en el rendimiento, con valores cercanos a 0,985. El algoritmo *AdaBoost* iguala el desempeño del *Decision Tree* en la configuración [5, 10] (F1-score de 0,98781), pero lo supera en otras combinaciones. Los modelos de las tablas 2 y 3 muestran además alta especificidad (mayor o igual a 0.984) y valores predictivos consistentes en los parámetros PP (valor predictivo positivo) y NP (valor predictivo negativo) con valores mayores o iguales a 0.98, aunque con mayor variabilidad en sensibilidad (rango: 0.960–1.000). Los resultados sugieren que valores intermedios de  $t$  (20–30) optimizan el balance entre sensibilidad y especificidad, superando el rendimiento de  $t = 50$  en la misma dimensión de *embedding*. La diferencia entre las métricas F1-train y F1-test mostró valores menores a  $\text{dif} - F1 < 0,05$  lo que indica que no hay sobre ajuste para los casos informados. A modo de ejemplo se muestran las curvas aprendizaje del caso AdaBoost  $[d, t] = [6, 20]$  en la figura 3.

D	K	Modelo	F1-Score	Balanced Acc	Sensitivity	Specificity	PP Values	NP Values	F1-train	F1-test	dif-F1	Overfitting
4	20	AdaBoost	1	1	1	1	1	1	1	1	0	NO
4	50	AdaBoost	0.99223	0.99231	1	0.98462	0.9849	0.98462	1	0.99223	0.00777	NO
4	30	AdaBoost	0.9894	0.98661	0.97321	1	1	1	1	0.9894	0.0106	NO
5	10	AdaBoost	0.98781	0.9875	0.975	1	1	1	1	0.98781	0.01219	NO
5	10	Decision Tree	0.98781	0.9875	0.975	1	1	1	1	0.98781	0.01219	NO
6	10	AdaBoost	0.98706	0.98798	0.99592	0.98004	0.97894	0.98004	1	0.98706	0.01294	NO
6	20	AdaBoost	0.98522	0.9856	0.98062	0.99059	0.98987	0.99059	0.99199	0.98522	0.00677	NO
6	30	AdaBoost	0.98416	0.98036	0.96073	1	1	1	0.99556	0.98416	0.0114	NO
5	30	AdaBoost	0.98364	0.98392	0.97895	0.98889	0.98947	0.98889	1	0.98364	0.01636	NO
5	20	AdaBoost	0.98359	0.98414	0.98462	0.98367	0.984	0.98367	1	0.98359	0.01641	NO

Tabla 2: Resultados de clasificación de los grupos NOR y DIA.



D	K	Modelo	F1-Score	Balanced Acc	Sensitivity	Specificity	PP Values	NP Values	F1-train	F1-test	dif-F1	Overfitting
4	10	AdaBoost	0.96366	0.96349	0.92698	1	1	1	1	0.96366	0.03634	NO
4	20	AdaBoost	1	1	1	1	1	1	1	1	0	NO
4	30	AdaBoost	0.9894	0.98661	0.97321	1	1	1	1	0.9894	0.0106	NO
4	50	AdaBoost	0.99223	0.99231	1	0.98462	0.9849	0.98462	1	0.99223	0.00777	NO

Tabla 3: Resultados de clasificación de los grupos NOR y DIA para el caso de dimensión de *embedding* 4.

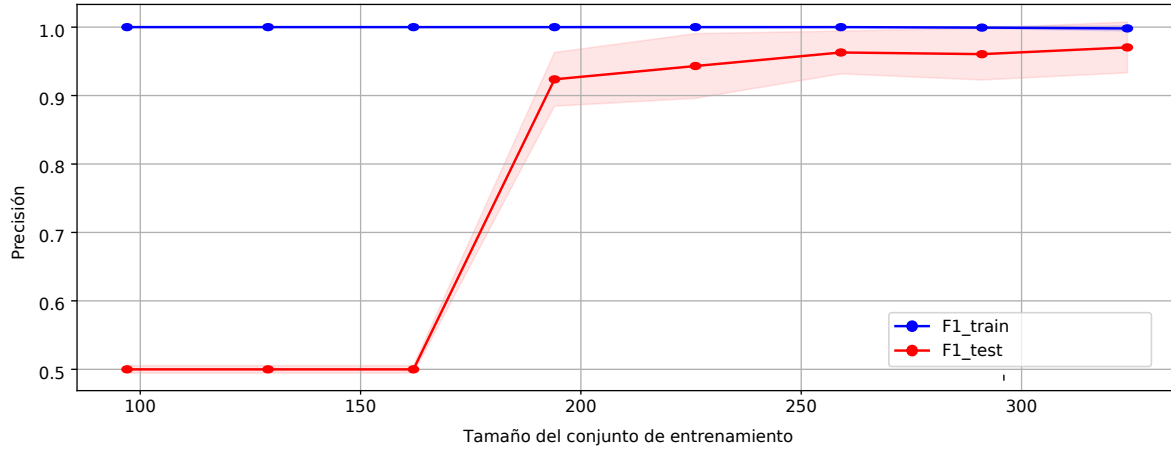


Figura 3: Curvas de aprendizaje correspondientes a la evolución de la precisión de entrenamiento y la precisión de testeo para el caso AdaBoost  $[d, t] = [6, 20]$ .

### 3. CONCLUSIONES Y DISCUSIÓN

El uso del modelo GFS junto al modelo AdaBoost resultó en una combinación exitosa en la clasificación de las señales de glucemia ya que alrededor del 13,75 % de las combinaciones  $([d, t], \text{ModeloML})$  calculadas arrojaron resultados por encima de 0,9 en sus valores de F1-score. En particular para los datos de entrada  $([4, \dots], \text{AdaBoost})$  se vio que hay una clasificación consistente en todos los casos analizados, lo que implica una coherencia en el mecanismo de estos modelos.

A pesar de que hay una diferencia sustantiva entre la cantidad de señales normales y diabéticas que puede dar lugar a un desbalanceo de los resultados dado que hay mucha más información sobre un grupo que sobre otro, este aspecto queda resuelto por el algoritmo GFS al quitar la información redundante. Esto queda evidenciado por los resultados mostrados en la tabla de 1 donde, incluso hay casos en los que las señales diabéticas contienen una mayor cantidad de vectores. Por este motivo resulta importante destacar que el algoritmo GFS fue capaz de compensar el desbalanceo presente en la base de datos utilizada.

Es posible que exista alguna configuración de hiperparámetros en algunos modelos descartados, que mejore los resultados y se espera que así ocurra. Eso requiere de un estudio detallado del uso del GFS en cada uno de los algoritmos ML. En esta primera instancia se pudo ver el comportamiento de los modelos ML, junto con el espacio GFS en la clasificación de las señales dadas encontrando que el modelo AdaBoost responde bien para todas las configuraciones iniciales  $[d, t]$ .

De manera general resultados obtenidos refuerzan la idea de que la combinación de aprendizaje automático con representaciones geométricas de esta clase de señales permite una clasifi-

cación eficiente de datos de glucemia, con AdaBoost sobresaliendo como un modelo especialmente efectivo en este contexto.

#### 4. TRABAJOS FUTUROS

La meta principal fue alcanzar la clasificación más alta posible. Se eligió el criterio de discretización basándose en este resultado, utilizando el factor de discretización más pequeño que mejora la clasificación. Por ende, todavía no se ha puesto en marcha una función de costo combinada ni una optimización de múltiples objetivos. Este aspecto se considera importante y será tratado en futuros trabajos.

En futuras etapas de este trabajo se plantea retomar el análisis del conjunto de datos provenientes del espacio GFS, poniendo especial atención en la relación entre dicho espacio intermedio y el espacio original de las señales de glucemia. El verdadero desafío no radica en el análisis dentro del espacio GFS, sino en comprender cómo la información representada en él se vincula con el dominio original, que es el que realmente interesa para el desarrollo de un sistema clasificador capaz de detectar casos de Diabetes Mellitus tipo 2.

El trabajo de investigación actual todavía está en proceso y un propósito importante es el de alcanzar una adecuada capacidad de generalización del modelo, para que el sistema sea capaz de reaccionar con eficacia ante señales de glucemia que no estén incluidas en el conjunto de entrenamiento. Si todas las señales de glucemia se utilizan en la construcción del espacio GFS, el algoritmo logra una excelente clasificación de las señales disponibles, lo que demuestra su capacidad de aprendizaje dentro del conjunto analizado. Sin embargo, al no haber señales externas al proceso de entrenamiento, el modelo no puede aún demostrar su capacidad de extrapolar y predecir correctamente ante señales nunca vistas. Por ello, será conveniente incorporar estrategias de validación más rigurosas que permitan evaluar y fortalecer su capacidad predictiva frente a nuevos datos.

#### REFERENCIAS

- Abhari S., Hassanzadeh H., y Jalili S. Artificial intelligence applications in type 2 diabetes mellitus care: Focus on machine learning methods. *Frontiers in Artificial Intelligence*, 6:1123450, 2023. <https://doi.org/10.3389/frai.2023.1123450>.
- American Diabetes Association. Standards of medical care in diabetes. 2023. Consultado en julio 2023.
- Carrell A.M., Mallinar N., Lucas J., y Nakkiran P. The calibration generalization gap. *arXiv preprint arXiv:2210.01964*, 2022.
- García Blesa M., Vorobioff J., y Legnani W. An innovative method to distinguish chaos from noise in the time domain. *International Journal of Applied and Computational Mathematics*, 11:161, 2025. <https://doi.org/10.1007/s40819-025-01991-x>. Published 04 August 2025.
- Genuth S.M., Palmer J.P., y Nathan D.M. Classification and diagnosis of diabetes. 2018.
- Géron A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc.", 2022.
- Herman W.H., Ye W., Griffin S.J., Simmons R.K., Davies M.J., Khunti K., Rutten G.E., Sandbaek A., Lauritzen T., Borch-Johnsen K., et al. Early detection and treatment of type 2 diabetes reduce cardiovascular morbidity and mortality: a simulation of the results of the anglo-danish-dutch study of intensive treatment in people with screen-detected diabetes in primary care (addition-europe). *Diabetes care*, 38(8):1449–1455, 2015.
- James G., Witten D., Hastie T., y Tibshirani R. *An Introduction to Statistical Learning with*



- Applications in R*. Springer Texts in Statistics. Springer, New York, 2nd edición, 2021.
- Kodama S., Fujihara K., Horikawa C., Ishiguro H., Yachi Y., Saito K., Shimano H., y Sone H. Predictive ability of current machine learning algorithms for type 2 diabetes mellitus: A meta-analysis. *Diabetologia*, 65(3):526–538, 2022. <https://doi.org/10.1007/s00125-021-05614-1>.
- Murphy K.P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- Rodriguez de Castro C., Vigil L., Vargas B., Garcia Delgado E., Garcia Carretero R., Ruiz-Galiana J., y Varela M. Glucose time series complexity as a predictor of type 2 diabetes. *Diabetes/metabolism research and reviews*, 33(2):e2831, 2017.
- Zihlmann M., Perekrestenko D., y Tschannen M. Convolutional recurrent neural networks for electrocardiogram classification. En *2017 Computing in Cardiology (CinC)*, páginas 1–4. 2017. <https://doi.org/10.22489/CinC.2017.070-060>.