ISSN 1870-4069

Clasificador de gestos motrices utilizando vectores de atributos distancia DTW sobre series de tiempo en representación SAX

Luis E. Valle[1], Rodolfo Romero[2], Jesús Y. Montiel[3]

[1] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
México

[2] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Laboratorio de Posgrado en Sistemas Computacionales Móviles,
México

[3] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Robótica y Mecatrónica,
México

lvalle212@gmail.com, rromeroh@ipn.mx, yalja@ipn.mx

Resumen. Este trabajo presenta una revisión del análisis de series de tiempo aplicado a la clasificación y reconocimiento de patrones. El artículo indaga en 3 métodos y compara su desempeño para la tarea específica del reconocimiento de gestos motrices empleando series de tiempo multivariables que derivan de la medición en la aceleración en los 3 ejes espaciales sobre un dispositivo sensor acelerómetro. Como aportación, se plantea un nuevo método capaz de superar la técnica de referencia y modelos del estado del arte con un 94.06 % de precisión en el conjunto de datos *GesturePeeble*; disponible en *The UCR Time Series Archive*. El método propuesto implementa el algoritmo *Dynamic Time Warping* para la conformación de vectores con atributos de distancia entre series de tiempo representadas con *Symbolic Aggregate Approximation*, posibilitando para las fases posteriores (entrenamiento y predicción) su ingreso en el modelo elegido. El método permite la elección del clasificador a conveniencia de la aplicación, optando para los resultados reportados en este trabajo, un modelo clásico de Máquinas de Soporte Vectorial.

Palabras clave: Clasificación de series de tiempo, deformación temporal dinámica (DTW), aproximación simbólica agregada (SAX).

Luis E. Valle, Rodolfo Romero, Jesús Y. Montiel

Motor Gesture Classification Using DTW-Based Feature Vectors on Time Series with SAX Representation

Abstract. This work features a review of the time series analysis applied to pattern classification and recognition. The paper deepens in 3 methods and compares their performance for the specific task of gesture recognition using multivariate time series derived from the measurement in the 3-spatial-axis acceleration on an accelerometer sensor device. As a contribution, a new method capable of surpassing the reference technique and state-of-the-art models with 94.06 % accuracy in the GesturePebble dataset is introduced; dataset available in the UCR Time Series Archive. The proposed method implements the Dynamic Time Warping algorithm for the composing of distance feature vectors between time series in Symbolic Aggregate Approximation representation, providing for the subsequent phases (training and prediction) their input in the chosen model. This method allows the classifier to be elected at the application's convenience, designating for the reported results in this work, a classical Support Vector Machine model.

Keywords: Time series classification, dynamic time warping, symbolic aggregate approximation.

1. Introducción

El reconocimiento de gestos es materia de interés en un amplia variedad de áreas del conocimiento, este trabajo se desarrolla en el marco específico de una propuesta que tiene como objetivo la creación una herramienta embebida que facilite la comunicación verbal para pacientes que recientemente han sufrido una lesión por traumatismo craneoencefálico o un accidente cerebrovascular que ha dañado principalmente el lóbulo izquierdo del cerebro.

El tipo de lesiones que derivan de accidentes de esta naturaleza generalmente ocasionan 3 condiciones médicas típicas: Las afasias[1], apraxias[2] y disartrias[3]. Los síntomas de estas condiciones se manifiestan como alteraciones del lenguaje, pérdida de la capacidad para la realización de gestos diestros aun cuando se mantiene el deseo y capacidad física de hacerlos, así como trastornos en la ejecución motora del habla.

La herramienta propuesta que permitirá al paciente conformar palabras replicando gestos de un código motriz, parte de la propuesta, atraviesa por una etapa de conformación de texto y finalmente de producción oral. Lo que compete a este artículo, en específico es la investigación de las técnicas y métodos disponibles para el desarrollo del algoritmo que se encargará de la clasificación de los gestos, como series dependientes del tiempo, a clases alfabéticas. Considerando el contexto al que aporta este trabajo, se toma especial consideración en la elección del conjunto de datos prueba y el desarrollo

experimental, esto con la motivación de replicar lo más fielmente las condiciones en las que funcionará la propuesta de herramienta.

Existen 2 decisiones especialmente importantes que se consideran para la elección del modelo de reconocimiento. La primera de estas es la naturaleza del código motriz, el cual en este caso será dependiente únicamente de las mediciones de un sensor acelerómetro. Esta decisión aporta grandes posibilidades al método de reconocimiento, como por ejemplo la posibilidad de intercambiarse de un prototipo de módulo sensor a dispositivos de uso común, tal como pueden ser relojes inteligentes y otros tipos de dispositivos *wearable*.

La segunda condición es la limitante en hardware disponible en un sistema embebido. Originalmente esta propuesta considera un entorno de RaspberryPi 4 para el algoritmo de reconocimiento, dispositivo en el que no es conveniente utilizar técnicas de aprendizaje profundo o algoritmos costosos en tiempo de ejecución computacional (exclusivamente en la etapa de predicción) motivación por el cual este trabajo busca alternativas que ofrezcen una alta precisión de predicción sin comprometer requerimientos de *hardware* no disponibles en un sistema en chip u ordenador en una placa.

2. Trabajos relacionados

Las series de tiempo, inherentemente descritas por una noción de ordenamiento en la que el tiempo es la unidad más típica, les mantiene constantemente presentes en casi cualquier tarea que involucra un proceso cognitivo humano[4], sucediendo a su vez en muchos otros fenómenos recurrentes en la naturaleza y cantidad de ámbitos de interés humano como negocios, economía, ingeniería, medio ambiente, medicina y otras áreas de investigación científica que recolectan datos en forma de secuencias dependientes del tiempo[5].

La tarea de clasificación de series temporales o *Time Series Classification*(TSC), consiste en el entrenamiento de un clasificador en un conjunto de entradas de entrenamiento en unos casos específicos donde cada uno contiene un conjunto ordenado de atributos en valores reales y una etiqueta de clase[6]. Se trata de un tema ampliamente estudiado y de gran interés en un extenso rango de áreas como *data mining*, estadística, procesamiento de señales, ciencias ambientales, biología computacional, procesamiento de imágenes, quimiometría, etc[6]. Representa tal importancia e influencia que cualquier problema de clasificación que utiliza datos registrados tomando en consideración una noción de ordenamiento puede ser transformado a un problema de TSC [4].

Con los años, investigadores en el tema han invertido un gran esfuerzo en estudios para el desarrollo de modelos que resuelvan el problema planteado por TSC y cada vez con una mayor precisión. Con la creciente disponibilidad de datos temporales[4] y archivos como *UCR Time Series Archive*[7], se ha propiciado el crecimiento en el número de algoritmos propuestos para TSC, implementando técnicas que utilizan modelos de aprendizaje máquina y diferentes medidas de distancia, técnicas para la transformaciones del espacio de los datos[6], técnicas de *ensembling*(entrenamiento de un conjunto de modelos menores que

integrando sus salidas resultan en un mejor modelo)[8], hasta nuevas alternativas de modelos de aprendizaje profundo en tendencia para muchas áreas de la IA, con una influencia relativamente nueva pero creciente en lo referente al problema de TSC[4].

Retomando el marco de este trabajo con la propuesta de herramienta de apoyo embebida como facilitadora de la comunicación verbal para personas con afecciones del habla, existen trabajos como [9,10,11,12], que atienden un problema similar pero dirigen su enfoque en la traducción de gestos pertenecientes a un lenguaje de señas, estas aplicaciones plantean una condición primordial, el previo conocimiento y dominio del lenguaje señado. Con gran diferencia la situación que enfrentan los pacientes que recientemente han sufrido de alguna lesión encefálica es distinta, no cumplen con la condición del dominio o conocimiento del lenguaje de señas, lo que es más, la mayoría contrario a eso y comprometido por el control pleno o parcial del lenguaje oral, desconocen un lenguaje complejo como el Lenguaje de Señas Mexicano(LSM), en el caso específico de México, sucediendo de la misma manera para gran porcentaje de la población en la sociedad que no convive en su entorno con una persona con afectaciones en el habla.

Aunado al complejo proceso de adquisición de un lenguaje señado, que equivale al aprendizaje de un nuevo idioma, se tienen los síntomas como la incapacidad para realizar gestos diestros y finos, como el movimiento controlado y preciso de los dedos para algunos gestos del lenguaje. La decisión de utilizar el acelerómetro permite entonces mantener en un mínimo el *hardware* sensor, puede portarse sin ser invasivo y es mucho más preciso para describir gestos amplios(movimiento de brazos y manos) más fácilmente ejecutables por pacientes con las afecciones descritas en[1,2,3].

Enfocándose principalmente en el objetivo que atiende este trabajo se distingue un artículo que atiende en profundidad, y de manera particular, la identificación de gestos motrices utilizando un sensor acelerómetro para el muestreo de las fuerzas en los 3 ejes durante la ejecución del patrón con una mano[13]. En el trabajo, Mezari y Maglogiannis, utilizan un *smartwatch* Pebble como elemento sensor, exponiendo el objetivo del trabajo como una examinación del uso de dispositivos básicos como *smartphones* y *wearables* para el reconocimiento confiable de gestos simples y naturales, proponiendo además una metodología para mejorar el desempeño y precisión. De la metodología elaborada por Mezari y Maglogiannis en su artículo[13] se toma parte de inspiración para desarrollar la propia del trabajo, siendo a la vez su método uno de los 3 evaluados, así como su principal componente en el método híbrido aportado por este artículo.

3. Descripción del conjunto de datos

Manteniendo en mente las condiciones del proyecto, se eligió acorde un conjunto de datos para el entrenamiento y prueba de los métodos experimentados. El conjunto de datos elegido *GesturePebble*, se deriva

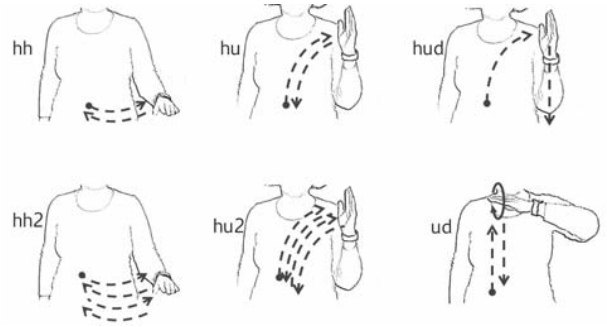

Fig. 1. Elección de los 6 gestos sencillos etiquetados y realizados por los participantes para la generación de las secuencias temporales disponibles en el *dataset GesturePebble*.

del trabajo *"Gestures Recognition using Symbolic Aggregate Aproximation and Dynamic Time Warping in Motion Data"*[13], el conjunto de datos se encuentra disponible para su uso público en el repositorio de series temporales *UCR Archive* [7].

El conjunto de datos resulta del estudio de dispositivos cotidianos como celulares y *wearables* para el reconocimiento de gestos, por lo que la recolección de las muestras se realizan con mediciones del acelerómetro del reloj inteligente Pebble en los 3 ejes espaciales mientras el dispositivo es utilizado en la muñeca por los participantes.

Incluye instancias de series temporales con longitud variable con un máximo de 456 mediciones por patrón. Los archivos provistos por UTC[7] colocan la etiqueta de clase como primer elemento de cada secuencia y aquellas series de tiempo de longitud menor han sido acompletadas con *NaNs*.

El manejo de los elementos *NaNs* se realiza en el código de la implementación al eliminarseles, esta acción otorga como resultado una serie de tiempo con la longitud de mediciones reales tomadas, más tarde esto no representa un inconveniente para los métodos de *Symbolic Aggregate Approximation*(SAX) y *Dynamic Time Warping*(DTW), ambos algoritmos manejan nativamente secuencias de longitudes variables y a su vez SAX procesa cada serie de forma individual.

El *dataset* involucra a 4 participantes, cada uno instruido para realizar 6 gestos(Figura 1) que repitieron durante un par de sesiones separadas entre días para conseguir un total de 8 grabaciones y completando el *dataset* a un total de 304 gestos.

De las series temporales captadas, se proveén las mediciones del sensor acelerómetro en el eje Z únicamente, facilitando a su vez un par de conjuntos de datos con características específicas cada uno: *GesturePebbleZ1* y *GesturePebbleZ2*.

El conjunto de datos *GesturPebbleZ1*, proveé un archivo con instancias para el entrenamiento del modelo clasificador, componiendose los patrones desarrollados

por los participantes durante la primera sesión. Mientras que el archivo de prueba, es la colección de los patrones realizados en la segunda sesión.

Para el conjunto *GesturePebbleZ2* el conjunto de entrenamiento consiste en los datos de un par de sujetos, mientras el conjunto de prueba se conforma del par de sujetos restante. Presumiblemente esta segunda colección está destinada a ser más compleja en dificultad que el primero, impidiendo que el modelo sea entrenado con patrones de los sujetos que más tarde serán evaluados. La dificultad de este *dataset* se argumenta basándose en el hecho de que cada participante posee una postura y mecánica de movimiento distinta a las de los demás.

4. Definición de algoritmos y técnicas utilizadas

4.1. *Dynamic Time Warping*

Dynamic Time Warping o DTW, es una técnica bien conocida para encontrar una alineación óptima entre 2 secuencias dependientes del tiempo(series de tiempo) dadas y bajo ciertas restricciones[14]. Este algoritmo es sumamente útil para medir la similaridad entre dos secuencias temporales que no se alinean exactamente en el tiempo, velocidad o extensión[14]. Originalmente este algoritmo había sido utilizado para comparar patrones del habla en reconocimiento de habla automático, siendo aplicado en otros campos de forma exitosa para lidiar con deformaciones en el tiempo y diferentes velocidades.

El objetivo de DTW es la comparación de 2 secuencias dependientes del tiempo X y Y, siendo series discretas, o más generalmente una secuencia de características muestreadas a puntos equidistantes en el tiempo.

$$\begin{aligned} X &= (x_1, x_2, ..., x_N) \;\; N \in \mathbb{N} \\ Y &= (x_1, x_2, ..., x_M) \; M \in \mathbb{N}. \end{aligned} \tag{1}$$

Con un *espacio de características* denotado por $\mathcal{F}$, entonces $x_n, y_m \in \mathcal{F}$ con $n \in [1 : N]$ y $m \in [1 : M]$. Para comparar dos características diferentes se realiza mediante una *medida de costo local*, también llamada *medida de distancia local* definida por:

$$c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}. \tag{2}$$

La implementación de la *medida de distancia local* depende de la medida de distancia general definida, para el documento se implementa la distancia Euclidiana c_e, definida como: $c_e(x_i, y_j) = (x_i - y_j)^2$ donde $i \in [1 : N]$ y $j \in [i : M]$, pero otro tipo de medidas de distancia local pueden utilizarse, como por ejemplo L1 o distancia *Manhattan*.

Típicamente *c(x,y)* es pequeña(bajo costo) si x y y son similares, de otra forma esta es alta. Evaluando el costo local medido para cada par de elementos de las secuencias X y Y, se obtiene la *matriz de costo* $C \in \mathcal{R}^{N \times M}$ definida por $C(n, m) := c(x_n, y_m)$ entonces la meta es encontrar una alineación entre X y Y obteniendo el costo total mínimo.

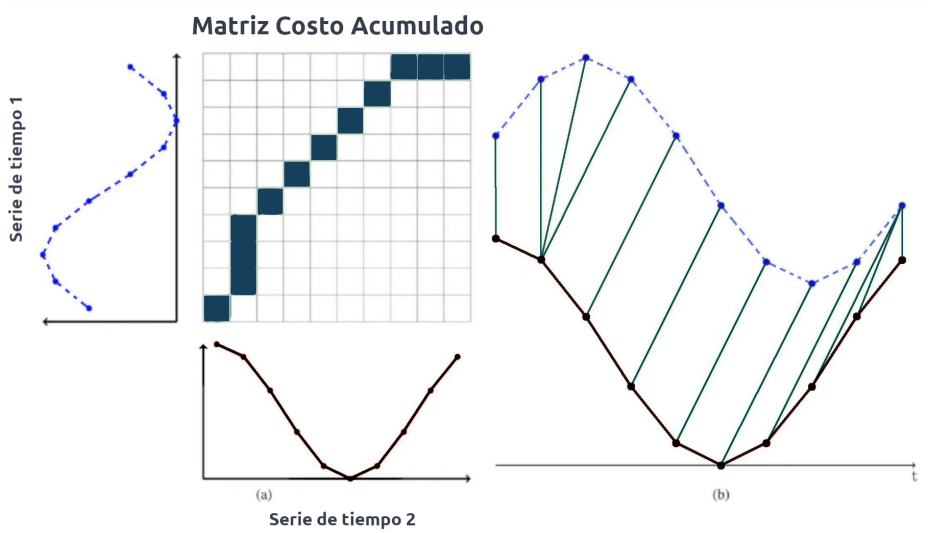

Fig. 2. Matriz de costo acumulada del cálculo de la distancia entre la Serie de tiempo 1 y Serie de tiempo 2.
a) Los recuadros rellenos en la cuadrícula conforman el camino de deformación óptimo(*optimal warping path*) p^* en la matriz de costo.
b) Resultado de la alineación óptima para la Serie de tiempo 1 en la Serie de tiempo 2 utilizando DTW.
Figura recuperada de: Thales Sehn Körting(Productor).(2017) *How DTW (Dynamic Time Warping) algorithm works*[YouTube]. https://www.youtube.com/watch?v=_K1OsqCicBY.

Formalmente un camino de deformación o *warping path*, es una secuencia $p = (p_1, ..., p_L)$ con $p_\ell = (n_\ell, m_\ell) \in [1:N] \times [1:M]$ para $\ell \in [1:L]$ satisfaciendo las siguientes 3 condiciones:

(I) *Condición límite*: $p_1 = (1,1)$ y $p_\ell = (N,M)$
(II) *Condición Monotonicidad*: $n_1 \leq n_2 \leq ... \leq n_L$ y $m_1 \leq m_2 \leq ... \leq m_L$
(III) *Condición del tamaño del paso*: $p_{\ell+1} - p_\ell \in (1,0),(0,1),(1,1)$ para $\ell \in [1 : L-1]$

Un *camino de deformación-(N,M)* $p = (p_1, ..., p_L)$ define una alineación entre las secuencias X y Y al asignar el elemento x_{n_ℓ} al elemento y_{m_ℓ}. Un ejemplo de la aplicación de las 3 condiciones durante la construcción del camino óptimo se observa en la Figura 2, donde utilizando la matriz de costo acumulado se marca una secuencia, con las casillas rellenas, de las distancias óptimas satisfaciendo siempre las condiciones(i,ii,iii).

El costo total $c_p(X,Y)$ de un camino de deformación p entre X y Y con respecto a la medida de costo local c se define como:

$$c_p(X,Y) = \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell}). \tag{3}$$

Mientras que un camino de deformación óptimo entre X y Y es un camino de deformación p^* que tiene un costo total mínimo entre todos los posibles caminos de deformación.

Finalmente, la distancia DTW entre X y Y es definido entonces como el costo total de p^*:

$$\begin{aligned} &DTW(X,Y) = c_{p^*}(X,Y), \\ &DTW(X,Y) = min\{c_p(X,Y)|p : CaminoDeformación - (N,M)\}. \end{aligned} \tag{4}$$

4.2. Banda Sakoe-Chiba

DTW es un algoritmo muy popular por sus excelentes resultados en la comparación de secuencias temporales, y sin embargo debido a la construcción de la matriz de costos, su complejidad cuadrática es un detractor significativo en tiempo y precisión de cálculo. Una variante común en DTW que aborda esta situación es la implementación de la banda Sakoe-Chiba[15].

De forma general, la implementación de restricciones a DTW aporta un par de beneficios. El primero de ellos y el más importante si se plantea el objetivo de la optimización, es la reducción de complejidad con la implementación de la banda, forzando al cálculo de un subconjunto del espacio global e inmediatamente reduciendo su complejidad de $O(L^2)$ a $O(w \times L)$ con $0 \geq w \leq L-1$, donde w define una banda.

El segundo de los beneficios refiere en cuanto a la fiabilidad de la similitud, pues previene alineaciones patológicas, evitando la desviación excesiva de la diagonal principal al camino óptimo posible.

La banda Sakoe-Chiba corre a lo largo de la diagonal principal y tiene un ancho fijo $T \in \mathbb{N}$ horizontal y verticalmente. Esta restricción implica que para un elemento x_n puede ser alineado únicamente a uno de los elementos de y_m con $m \in \left[\frac{M-T}{N-T} \times (n-T), \frac{M-T}{N-T} \times n + T\right] \cap [1:M]$. En la matriz de costo acumulado la ventana de deformación(*warping window* o WW) se observa como en la Figura 3.

4.3. Optimización del tamaño de la ventana de DTW

El algoritmo DTW es un método robusto y con una extraordinaria competitividad, además de utilidad, probando ser una medida de distancia para series de tiempo excepcionalmente fuerte, razón por la que es de gran importancia para la comunidad el conocimiento de la capacidad que logra el algoritmo cuando se optimiza correctamente el tamaño de la ventana w, la cual es crítica en la disminución del error durante la predicción[16].

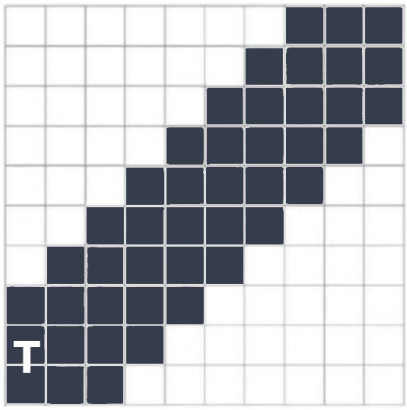

Fig. 3. Banda de Sakoe-Chiba que corre sobre la diagonal principal de la matriz y que tiene un ancho fijo T.

Argumentando que la restricción en su cantidad máxima de deformación, cuando es establecida correctamente, cierra la mayoría de los espacios de mejora obtenidos con métodos de clasificación de series de tiempo "más sofisticados". El artículo[17] propone un método nobel para el aprendizaje del parámetro óptimo en configuraciones supervisadas y no supervisadas, siendo el caso primero el que compete a este trabajo.

Existe una concepción errónea de que el valor de tamaño de ventana w puede ser dependiente del dominio de cálculo requiriendo su cálculo por única vez, cuando en realidad no existe un solo valor de w que pueda ser transferible entre diferentes contextos. El valor óptimo para la banda restrictiva de DTW dependerá de 2 factores, el número de instancias del conjunto de datos y la estructura propia de los datos, contundentemente abatiendo la posibilidad de la existencia de un valor prototípico de w para aplicaciones específicas como ritmos cardíacos o gestos[17].

Formalmente el problema se plantea como: Dado un conjunto de entrenamiento de series de tiempo etiquetadas, encontrar el valor de w(Tamaño de ventana Sakoe-Chiba) que maximiza la calidad de clasificación en un conjunto de prueba sin etiquetar.

La evaluación de la calidad de clasificación se realiza mediante la medida de la precisión.

Debido al creciente consenso que ubica al método DTW-k-NN como una línea base robusta, deciden los autores del trabajo[17] utilizar el algoritmo k-NN como el clasificador subyacente. Como se describe en secciones posteriores, este método es utilizado en todos los modelos evaluados, de forma que clasificadores como k-NN y SVM son implementados también con diferencia al artículo original[17].

El enfoque que toma este método es particularmente útil cuando el conjunto de entrenamiento es limitado, pues implementa una técnica de remuestreo con

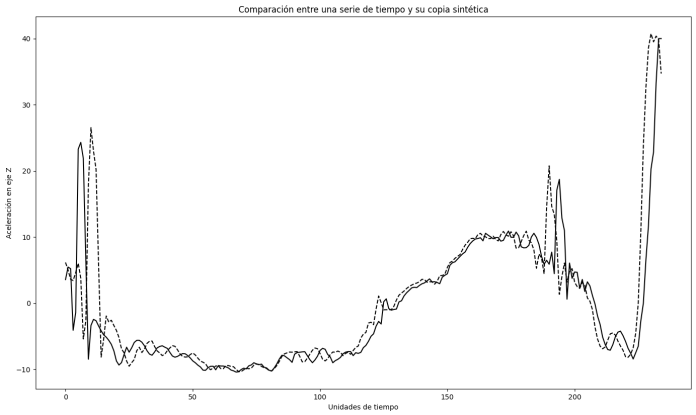

Fig. 4. Comparación de una instancia perteneciente al conjunto de datos *GesturePebble*(línea continua) y su remplazo sintético(línea punteada).

la creación de datos sintéticos que reemplazan al problema de un número de instancias limitadas.

El algoritmo consiste en hacer N copias del conjunto de entrenamiento original, reemplazando para cada copia una fracción de los datos con reemplazos sintéticos, para posteriormente realizar validación cruzada con el fin de aprender el porcentaje de error contra la curva de valores de w. Se utiliza el valor promedio de todas las iteraciones N para la predicción del mejor tamaño de ventana w.

De los componentes lógicos que conforman al método, los de mayor interés son el proceso de creación de un nuevo conjunto de datos con una porción sintética, y la función capaz de otorgar una deformación a una secuencia para otorgar un dato sintético.

El proceso de creación de un nuevo conjunto de datos con instancias sintéticas inicia por dividir aleatoriamente el conjunto de datos cumpliendo con una relación especificada de porcentaje de datos sintéticos sustitutos, mientras la otra partición permanece sin modificación. Posteriormente utilizando *K-Fold Cross Validation*, se realiza la división en conjunto de entrenamiento y prueba para medir iterativamente la calidad del clasificador.

La función capaz de sintetizar una serie de tiempo inicia con el encogimiento no lineal de la medición real mediante la eliminación aleatoria de un porcentaje definido por el usuario, de los datos que la componen. Seguido a esto, utilizando una función común a bibliotecas de procesamiento de señales, la nueva serie debe muestrearse una vez más a la misma longitud de la secuencia original, agregando antes de este proceso, un acolchado(*padding*) de la señal repitiendo en los extremos los últimos y primeros 10 valores respectivamente. Finalmente y terminado el remuestreo, se eliminan los valores de acolchado en los extremos y se obtiene una serie sintética de longitud igual a la original.(Figura 4)

Esta técnica requiere la configuración de 3 parámetros para su funcionamiento, utilizando como valores constantes para la experimentación los

valores propuestos por los mismo autores del artículo[17], 20 % de deformación para la creación de datos sintéticos, una relación de 8 a 2 para la creación del nuevo conjunto de datos con copias sintéticas como número de instancias mayoritarias, y finalmente para el número de iteraciones *N*, mientras mayor sea su valor mejor, considerándose de forma conservadora en 10.

4.4. *Piecewise Aggregate Approximation*

Piecewise Aggregate Approximation o PPA, es un algoritmo que tiene como idea básica la reducción dimensional de una serie de tiempo de entrada mediante la partición de esta en segmentos del mismo tamaño, sobre cada cual se realiza el cálculo promedio de los valores en el segmento[18]. Con una serie de tiempo $Y = Y_1, Y_2, ..., Y_n$ con tamaño $n \in \mathbb{R}$ la partición o reducción a una serie $X = X_1, X_2, ..., X_m$ donde $m \leq n$, la ecuación que describe los elementos en la serie reducida es:

$$\overline{X_i} = \frac{m}{n} \sum_{j=\frac{n}{N(i-1)+1}}^{\frac{n}{M}i} x_j. \tag{5}$$

Es importante resaltar que antes de realizar la aproximación el vector debe ser z-normalizado, y una vez haya sido estandarizado la aproximación por partes se calcula.

Se identifican 2 casos durante la separación en segmentos. El caso trivial, $m < n$ y m es múltiplo de n, se trata dividiendo el vector en ventanas de tamaños iguales y realizando el cálculo del promedio para la asignación del nuevo valor por segmento. El segundo cuando $m < n$ y m no es múltiplo de n, no existe el balance para la división exacta por lo que es requerido un redimensionado de cada ventana que permita a cada elemento de la serie de salida ser el promedio de un segmento de mismo tamaño al vector de entrada[18].

Se observa un ejemplo en la Figura 5 de la discretización de un par de series de tiempo con pocos atributos pero de extensión distinta, que después de la transformación PAA son equiparables en longitud con un número de palabras *m=9*.

4.5. *Symbolic Aggregate Approximation*

Symbolic Aggregate Approximation o SAX, es una técnica desarrollada y enfocada en la reducción dimensional de una serie numérica con una serie de tiempo, a un espacio simbólico de 'palabras'. Dada una serie dependiente del tiempo de longitud arbitraria n se realiza la transformación a una cadena de longitud w' utilizando un alfabeto $A = a_1, a_2, ..., a_3$[19].

La primer parte de la discretización es manejada por la transformación PAA, siguiendole el proceso de asignación de símbolos para cada sección, conformando como requisito para esta segunda parte de la discretización que los símbolos sean asignados equi probablemente (propiedad de una colección de eventos teniendo

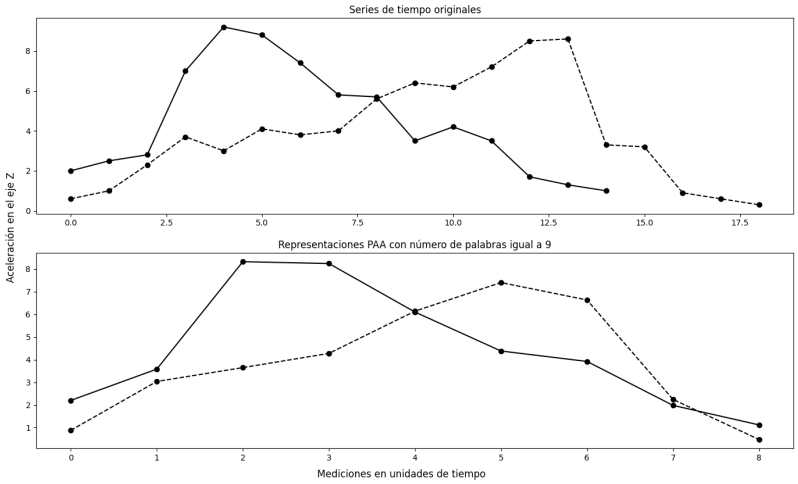

Fig. 5. El algoritmo PAA mantienen las tendencias de las secuencias.

la misma probabilidad de ocurrir). Esto se cumple fácilmente por la previa normalización de los datos de la serie en una distribución Gausiana, razón por la que las áreas bajo la curva de campana de la distribución pueden ser utilizadas para la creación de los puntos de quiebre(*breakpoints*) en los datos normalizados para la asignación de palabras.

El método algorítmico toma los siguientes procesos en orden [13] (Figura 6):

1. Estandarización o normalización-z de los datos de la serie de tiempo para media $\mu = 0$ y desviación estándar $\sigma = 1$.
2. Transformación o reducción de dimensionalidad desde n a w' mediante el algoritmo *Piecewise Aggregation Approximation*(PAA).
3. Asignación de los puntos de quiebre $\beta = \beta_1, ..., \beta_{\alpha-1}$.
4. La serie de tiempo es discretizada tomando el promedio de cada segmento para ser mapeado a un alfabeto A.

La distancia entre 2 palabras o cadenas, correspondiendo cada una a diferentes series de tiempo, se calcula como el promedio de los pares de las distancias de símbolos.

Los puntos de quiebre es una lista de números $\beta = \beta_1, ..., \beta_{\alpha-1}$ tal que el área debajo la curva gausiana de β_i a $\beta_{i+1} = \frac{1}{\alpha}$[19].

Estos puntos de quiebre pueden ser obtenidos mirando una tabla estadística y puede ser utilizada para la discretización de series de tiempo donde el coeficientes debajo el punto de quiebre más pequeño son mapeados a la letra del alfabeto en el índice 1, mientras los coeficientes mayores o igual al punto de quiebre más pequeño y menor al segundo punto de quiebre se le asigna la letra del alfabeto con el segundo índice, y así sucesivamente.

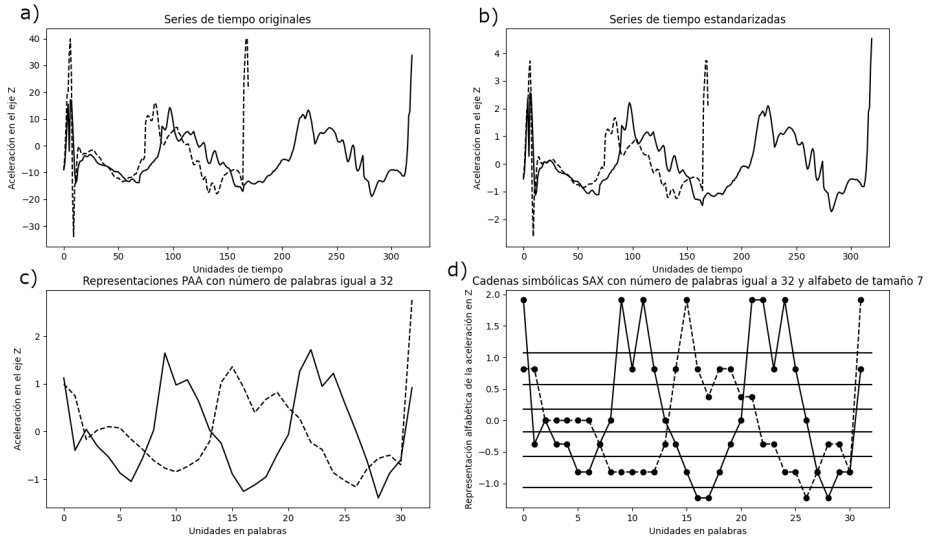

Fig. 6. Graficación de un par de series de tiempo tomadas del conjunto de datos *GesturePebble* durante el proceso completo de la discretización SAX
a) Series de tiempo de longitud variables con valores reales(originales)
b) Series de tiempo después de estandarización.
c) Series de tiempo después de la transformación PAA con un número de palabras igual a 32.
d) Series de tiempo en su representación SAX como una cadena de símbolos numéricos con alfabeto de tamaño 7.

La distancia entre 2 símbolos se define a 0 si el índice difiere a lo más por 1(por ejemplo resulta 0 entre símbolos a_i y a_{i+1}), en cualquier otro caso la distancia entre símbolos a_i y a_k, donde $k > i$, se define como $b_{k-1} - b_i$[13].

La distancia entre 2 cadenas correspondientes a diferentes series de tiempo, se calcula como el promedio de las distancias de los símbolos por parejas(por ejemplo; el promedio de la distancia entre el primer símbolo de cada serie, la distancia entre el segundo símbolo de cada serie, y así sucesivamente)[13].

4.6. *DTW Barycenter Averaging*

DTW Barycenter Averaging o DBA, es un algoritmo iterativo que utiliza DTW para la alineación de series de tiempo utilizando un promedio envolvente[20]. Introducido por Petitjean[20], la implementación del algoritmo DBA trae consigo varias ventajas importantes frente a un proceso plano o simple de promediación de un conjunto de secuencias.

El algoritmo *grosso modo*, opera de la siguiente manera:

1. Las n series por ser promediadas son etiquetadas $S_1, S_2, ..., S_n$ y tienen una longitud T.

2. Se inicia el proceso con una serie abreviada inicial *I*.
3. Se realizan los siguientes hasta que el promedio converge:
   a) Por cada serie *S*, se aplica DTW contra *I* y se salva el camino de deformación.
   b) Se utiliza el camino de deformación y se construye un nuevo promedio de *I* al dar a cada punto un nuevo valor: El promedio de cada punto de *S* conectado a este en el camino de deformación resultante de DTW.

Una buena inicialización del proceso con un candidato correcto para *I* es de extrema importancia, porque mientras el proceso DBA por sí mismo es determinístico el resultado final dependerá esencialmente de la secuencia abreviada inicial. Se identifican entonces 3 objetivos distintivos:

– Preservar la forma de las entradas.
– Preservar la magnitud de los extremos en el eje y.
– Preservar la sincronización de aquellos extremos en el eje x.

En el trabajo[20] se recomienda inicializar DBA eligiendo de forma aleatoria la serie abreviada de inicio *I*, y en un principio esta técnica preserva bien la forma, pero la sincronización de los extremos dependerá en cual serie resulta escogida, razón por la que un proceso determinístico de elección es preferible.

Un ejemplo de la aplicación de este algoritmo en el modelo propuesto se muestra en la figura 7, donde utilizando un conjunto de series de la misma clase se calcula una serie de tiempo representante de toda la clase.

5. Métodos de clasificación

5.1. DTW 1-NN

Un modelo obligatorio como punto de partida entre las vastas alternativas de técnicas que atienden el problema TSC, es a través de una combinación del algoritmo DTW y el modelo *k-NN* para la clasificación de secuencias dependientes del tiempo, mostrando sus capacidades de precisión y robustez cuando se compara con técnicas más poderosas[21].

Esta robusta técnica combina la medida de similitud entre secuencias de tiempo DTW y el algoritmo de clasificación k-NN(*k Nearest Neighbours*) en su variante con *k=1*, parámetro definido de forma empírica por infinidad de trabajos anteriores y para conformar actualmente el método base en el problema de TSC por la creciente aceptación por parte de la comunidad.

En esta variante del algoritmo k-NN, se sustituye la distancia Euclidiana como función de medida entre los datos por la medida de similitud DTW entre series de tiempo, resultando en su gran capacidad de clasificación con una buena precisión. Aún con los beneficios que otorga el modelo referencia, es importante revisar los detrimentos de la técnica frente a las limitaciones y condicionantes naturales de la problemática que se atiende. La gran debilidad de este algoritmo reside en la complejidad y tiempo de cálculo, dado principalmente por el uso de DTW como función de medida en el modelo clasificador.

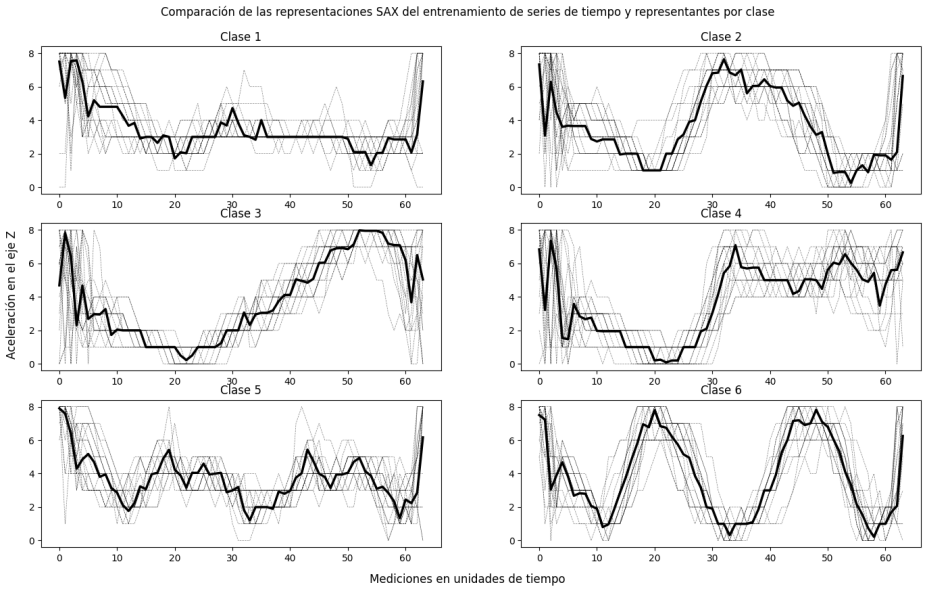

Fig. 7. Ejemplo de las series representantes de cada clase para el conjunto de datos *GesturePebble*. Cada representante(línea gruesa) fue calculada utilizando el algoritmo DBA con un promediado de 5 instancias discretizadas(línea delgadada y punteada) en representación SAX de alfabeto tamaño 9 y número de palabras 64.

La complejidad cuadrática de DTW, $O(N^2)$, cuando se evalúa la similitud de la series de tiempo que se intenta predecir con respecto al número de instancias que conforman el conjunto de datos de "entrenamiento", incrementa significativamente el número de cálculos derivado de la conformación de la matriz de costo del algoritmo. Frente a esta situación, se ha buscado la optimización del algoritmo utilizando las restricciones, en un intento de disminuir la complejidad en la conformación de la matriz de costo acumulada, inevitablemente requiriendo el aprendizaje del tamaño óptimo del parámetro tamaño de ventana w que restringe los cálculos a elementos más cercanos a la diagonal.

La versión desarrollada experimentalmente en este trabajo, implementa la técnica de optimización del tamaño de la ventana DTW[17] en la búsqueda de satisfacer las condiciones de la solución que exigen un modelo optimizado sin dejar de lado una buena precisión en la predicción.

5.2. Modelo SAX-DTW

Técnica propuesta en el artículo *"Gesture recognition using Symbolic Aggregate Approximation and Dynamic Time Warping in Motion Data"*[13], como el método más preciso de entre los 3 evaluados en aquel trabajo, supera a los demás al reportar una razón promedio de clasificación correcta igual al 99.21 %.

El método se basa en la coexistencia del algoritmo SAX y la función de medida DTW, logrando combinar las mejores características de ambos métodos; La efectividad y baja complejidad de SAX, con la insensibilidad de DTW a las fluctuaciones de velocidad durante la ejecución de un gesto[13].

Este segundo método, al igual que el de referencia, utiliza como clasificador subyacente a 1-NN, pero implementado como función de distancia la propia distancia SAX modificada para implementar la alineación óptima de DTW.

La totalidad de las secuencias temporales serán discretizadas a la representación SAX, tanto las instancias de conjunto de "entrenamiento", como los nuevos patrones ingresados para su etiquetado mediante predicción.

Importante para el método SAX aclarar los valores de los parámetros utilizados. En el trabajo desarrollan sus experimentos con un valor para el número de palabras igual a 32(Representación PAA) y un tamaño de alfabeto igual a 7.

Finalmente la función de medida adopta el concepto de la distancia entre 2 símbolos SAX con una modificación. En lugar de ocuparse la comparación por defecto donde 2 símbolos del mismo índice en diferentes cadenas se comparan(una comparación de distancia Euclidiana en el ámbito discreto simbólico de SAX), se ocupa el camino óptimo de deformación proponiendo una comparación de la distancia entre símbolos relacionados por el mismo camino óptimo de deformación DTW. En la Figura 8 se ilustra el uso del método SAX-DTW.

La implementación del algoritmo DTW en este método exige la optimización del parámetro tamaño de ventana w como restricción en forma de banda Sakoe-Chiba, aplicándose también durante los resultados de la experimentación el método de aprendizaje del tamaño óptimo de ventana[17].

5.3. Método de vectores de atributos distancia DTW

El trabajo del artículo *"Using dynamic time warping distances as features for improved time series classification"*[22] basa su metodología partiendo de la robustez de DTW como una medición de distancia para series de tiempo, y tiene como objetivo aprovechar tal fortaleza utilizando indirectamente DTW para la creación de nuevas características que pueden ser alimentadas posteriormente a un método de aprendizaje máquina estándar en vez de utilizarse con el típico 1-NN(*1 Nearest Neighbour*).

El autor Kate en su trabajo[22] describe la representación de las series de tiempo en términos de sus distancias DTW de entre cada uno de los ejemplos de entrenamiento, de tal forma que dada una serie de tiempo T y un conjunto de entrenamiento $D = [Q_1, Q_2, ..., Q_n]$ el vector de características resultante $Feature - DTW(T)$ de T construido utilizando DTW es simplemente:

$$DTW(T) = (DTW(T, Q_1), DTW(T, Q_2), ..., DTW(T, Q_n)). \quad (6)$$

Una de las grandes fortalezas que destaca el autor sobre su método, es la mejora en el desempeño gracias a la implementación de algoritmos más

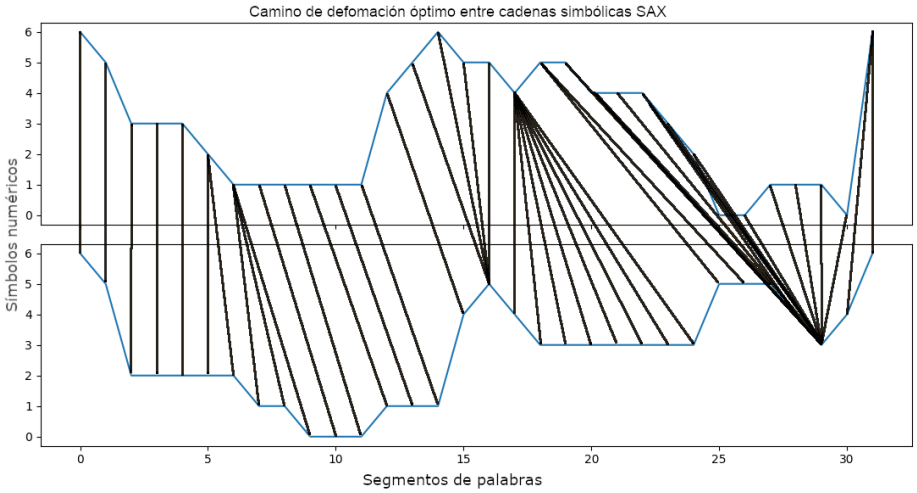

Fig. 8. Algoritmo del camino de deformación óptima calculada por DTW sobre 2 cadenas simbólicas SAX.

avanzados de aprendizaje máquina, eligiendo para la experimentación en su trabajo un modelo de Máquinas de Soporte Vectorial(SVM), indicando que de esta manera su método es capaz de aprender como la clase de una serie de tiempo se relaciona con sus distancias DTW de entre varios ejemplos de entrenamiento[22].

El método se distingue por ser además fácilmente extendible al utilizarse en combinación con otros métodos basados en características estadísticas y simbólicas añadiendolas simplemente como características adicionales. En el mismo trabajo Kate[22] desarrolla una variante de su método que concatena las características distancias DTW con características bolsa de palabras de SAX, reportándose en sus experimentos como el mejor clasificador no ensamble.

5.4. Método vectores de atributos distancia DTW de series de tiempo SAX

También nombrado Atributos SAX-DTW o *SAX-DTW Features*, es el método original que se propone en este trabajo, y que busca al igual que el modelo SAX-DTW, la coexistencia y adición de los beneficios que ofrece la disminución dimensional y discretización por parte de las representaciones SAX, con la robustez en el proceso de medición de la similitud de DTW para series con extensión y velocidades distintas. Aún cuando ambos algoritmos se implementan en un mismo modelo, el enfoque y filosofía de funcionamiento del modelo difiere de SAX-DTW, implementado una técnica desarrollada en el artículo[22] que propone representar a una serie de tiempo como un vector conformado por atributos calculados con las distancias DTW con respecto a cada ejemplo de entrenamiento[22].

Así como este método toma inspiración de las técnicas desarrolladas en ambos artículos[13,22], ninguna de las técnicas se aplica directamente como son descrita por los autores en sus respectivos trabajos, esto por que su combinación requiere realizar modificaciones para cumplir con el objetivo y resultados esperados del método nuevo.

Un cambio especialmente importante en comparación al par de métodos del que toma inspiración, es la implementación de un algoritmo *Machine Learning* mucho más robusto que el sencillo k-NN. Se ocupa en cambio un clasificador de soporte vectorial a raíz de ser considerado una de las grandes fortalezas cuando se ocupa un modelo clasificador más sofisticado con los nuevos vectores conformados por las distancias DTW como atributos[22] en comparación del sencillo k-NN.

Fase de preprocesamiento de los datos La manera más sencilla de obtener una mejora en la precisión de la clasificación para secuencias dependientes del tiempo es la implementación de un método basado en características(Representación estadística o simbólica definida para una serie de tiempo), y aprovechando este hecho se requiere la transformación de los datos en un espacio alternativo donde las características discriminatorias pueden ser más fácilmente detectadas que si se compara con un clasificador más complejo que permanece operando en el contexto temporal[6].

El manejo del total de series de tiempo, tanto las instancias que se utilizarán para entrenar el modelo clasificador como las que se alimentarán para predecir su etiqueta, es a través de su representación en cadenas simbólicas SAX, que proveé además de una transformación a un espacio alternativo de dominio simbólico, una reducción dimensional, lo que traduce en una simplificación de la complejidad. Esto implica que ninguna de las etapas siguientes pueda realizarse sin antes transformar las secuencias en cadenas simbólicas con número de palabras iguales a 64 y su discretización en el dominio de valores con un alfabeto tamaño 9.

Pensando en aprovechar todo el potencial de las similitudes calculadas por DTW, se escoge un alfabeto numérico $A \in [0, 8]$ en las representaciones SAX, permitiendo a su vez utilizar sin realizar una previa modificación a los símbolos, la distancia DTW.

Fase de entrenamiento (Figura 9) Cuando se involucra un método de aprendizaje máquina como SVC, el modelo debe pasar por una etapa de entrenamiento antes de poder ofrecer la tarea de predicción, situación que no sucede con el algoritmo k-NN de los métodos previos, pues realmente la etapa de entrenamiento y predicción sucede en un proceso integral.

La creación de los vectores de atributos en este método es esencial, no solo por la posibilidad que ofrece de aplicarse como entrada a modelos robustos de *Machine Learning*, pero además diverge del trabajo[22] en el que se inspira y se toma una interpretación alternativa que impacta positivamente en las etapa de entrenamiento y predicción, pero siendo de mayor relevancia su ventaja durante la predicción; Disminuyendo significativamente el tiempo de cálculo

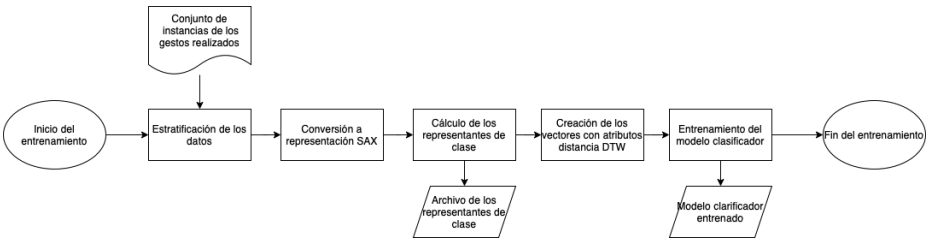

Fig. 9. Diagrama del algoritmo de entrenamiento para el modelo Atributos SAX-DTW

y complejidad requerido para la creación del vector de instancias por cada secuencia temporal.

Tomando un enfoque alternativo a la creación del vector de atributos, antes de pasar siquiera a esta etapa, se requiere crear una cadena simbólica representante de cada clase existente en el conjunto de datos, y como requisito se exige la estratificación del conjunto para mantener la equiprobabilidad de las clases durante el entrenamiento.

La fabricación del conjunto de cadenas representantes se logra mediante el promedio de todos los vectores existentes en el conjunto de datos que pertenecen a una misma clase, resultando el método más efectivo para lograr un representante fidedigno de cada etiqueta el algoritmo DBA[20].

Este conjunto de representantes resultante se convierte a partir de su cálculo en el conjunto de cadenas simbólicas más importantes del modelo. El hecho de que mediante la similitud DTW de estos representantes con los vectores destinados al entrenamiento y posteriormente los nuevos patrones a ser predecidos, exige la preservación del conjunto durante la vida útil del modelo entrenado. A pesar de tratarse de un algoritmo determinístico que sugiere el recálculo del conjunto de representantes frente a la pérdida de este, si la elección de la serie de tiempo abreviada inicial I se obtiene por un proceso aleatorio, un segundo cálculo del conjunto de representantes significa desechar el modelo entrenado para volver a crear las instancias de entrenamiento con este nuevo conjunto.

Creado el conjunto de cadenas representantes de cada clase, se calculan los vectores de atributos distancia DTW entre la instancia de entrenamiento y los representantes de clase(es importante el orden en que se ingresan ambas cadenas simbólicas, pues la operación de similitud DTW no es conmutativa), alimentando el modelo SCV para iniciar la etapa de entrenamiento.

Fase de predicción (Figura 10) Muestreado, filtrado y preprocesado un patrón de movimiento, el primer paso será representarlo mediante SAX en una cadena simbólica con el alfabeto numérico definido previamente A.

Como cadena simbólica, ahora esta secuencia puede ser usada para conformar un vector de atributos distancia DTW utilizando el mismo conjunto de representantes de clase usado para fabricar los vectores de entrenamiento.

Finalmente el vector resultante puede ser alimentado al modelo SVC previamente entrenado para la obtención de la etiqueta de clase resultante del proceso de predicción.

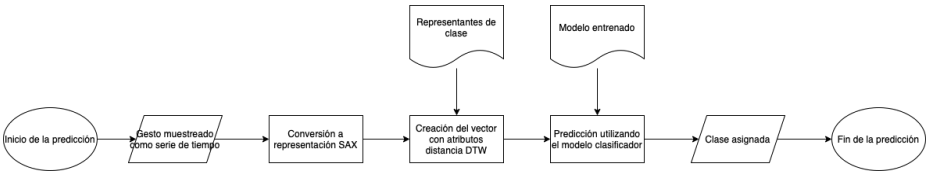

Fig. 10. Diagrama del algoritmo del proceso de predicción para el modelo Atributos SAX-DTW.

6. Resultados experimentales

El proceso experimental consiste en la replicación de los modelos descritos teóricamente en la sección previa, exceptuando el método de vectores de atributos distancia DTW, aplicando inicialmente la técnica de optimización para el tamaño de la ventana w para cada uno de los modelos y posteriormente midiendo la precisión del modelo utilizando el óptimo valor de w para el entrenamiento y fase de predicción con el conjunto de datos completo *GesturePebbleZ1*.

Se decidió por Python como el lenguaje de programación utilizado para la implementación y prueba de los modelos por su gran popularidad, sencillez de implementación y la gran disponibilidad de bibliotecas relacionadas con modelos de *Machine Learning* y especializadas en la Clasificación de Series de Tiempo. Aún con la variedad de bibliotecas existente la gran mayoría de algoritmos descritos fueron desarrollados, tomando únicamente como referencia los existentes en bibliotecas.

De los algoritmos utilizados en el trabajo, aquellos que son implementados desde bibliotecas y no se desarrollaron directamente, son el algoritmo para el muestreo de las señales en el proceso de optimización de la ventana DTW con la biblioteca *resampy*, y la función DBA la cual forma parte de la biblioteca nombrada *tslearn*. Igualmente a lo que refiere a los modelos clasificadores se implementaron funciones parte de las bibliotecas desarrolladas encargadas de ejecutar el sencillo algoritmo 1-NN con las variaciones necesarias para la adición de las diferentes funciones de medición con el modelo referencia y el modelo propuesto en el trabajo de reconocimiento de gestos utilizando dispositivos comunes[13]. La excepción se encuentra en el modelo SVC, para el cual se utilizó la función que ofrece scikit-learn y que requiere únicamente el ingreso de los conjuntos de entrenamiento, conjunto de clases y especificación de parámetros en la fase de entrenamiento, finalizando con la etapa de predicción donde es necesario alimentar el vector por predecir en el modelo entrenado.

Tabla 1. Tabla que muestra los resultados obtenidos para cada método con la técnica de optimización del tamaño de ventana w.

Optimización del tamaño de ventana w .

Métodos	Precisión	Tiempo(min)	Valor w
DTW 1-NN	85.83 %	25.28	**16**
SAX-DTW	89.83 %	6.28	**2**
Atributos SAX-DTW	**95.25** %	43.45	**4**

En este trabajo aunque se aborda teóricamente el modelo de del trabajo[22], no se reporta experimentalmente en esta sección. La razón principal de esta decisión es la inconsistencia mostrada en la implementación lograda, y que otorga resultados de entre 33.33 % y 50 % de precisión, muy por debajo del margen mínimo que se alcanza con los otros métodos.

Las respectivas implementaciones del algoritmo DTW, como función de medida o parámetro de caracterización, son evaluados previo aprendizaje del valor óptimo de la ventana para el contexto específico de cada modelo, tal como expone el artículo[17] no existe un valor único de w intercambiable entre contextos, con factores que afectan a su universalidad como la forma de los datos y el tamaño del conjunto.

Esta optimización del tamaño de ventana se realizó utilizando el conjunto de datos *GesturePeeble*, del que también se dispuso durante el entrenamiento y predicción final de los 3 modelos. Es requisito para resultados fidedignos utilizar un conjunto de datos estratificado, razón por la que el conjunto pasó de 132 entradas para 6 etiquetas distintas, a un total de 120 entradas con 20 secuencias temporales por cada clase. Los demás parámetros que se consideran comunes para todos los modelos con el número de iteraciones N igual a 10 y *k-Fold Cross Validation* igual a 10.

Los límites inferiores y superiores del cálculo de w si cambian en función del modelo y se definieron después de realizarse una iteración de prueba con un limite superior relativamente alto para identificar el rango de valores donde el modelo aumenta su precisión, evitando así el cálculo ocioso de un dominio exagerado de valores w. Con la intención de permitir la replicación de los resultados, se reportan los límites utilizados en la función que calcula el valor óptimo de la ventana DTW como: entre $[11, 17]$ para DTW 1-NN, entre $[2, 8]$ para SAX-DTW y finalmente entre $[2, 10]$ para vectores de atributos distancia SAX-DTW.

En el Cuadro 1 se reportan los resultados obtenidos de la técnica de optimización del valor de la ventana w, siendo importante rescatar el reducido tamaño obtenido para la ventana en el par de métodos que incluyen una transformación de espacio a un dominio simbólico mediante SAX. La sola reducción del valor de w ya es favorable en una evaluación que traduzca el número de operaciones en relación al tamaño de su ventana para series de tiempo hipotéticamente de longitud similar, y más sin embargo la situación es distinta

Tabla 2. Valores de los parámetros utilizados para la evaluación del desempeño de los diferentes métodos.

Parámetros	**Valores**
Tamaño *dataset* entrenamiento	120
Tamaño *dataset* prueba	150
Número de corridas	10
Tamaño de ventana w	Aprendido para cada método

Tabla 3. Resultados promedio obtenidos en la precisión de la predicción para cada método.

Precisión en la predicción		
Métodos	Precisión	Tiempo(seg)
DTW 1-NN	75.20 %	57.91
SAX-DTW	90.33 %	12.89
Atributos SAX-DTW	**94.06** %	**5.181**

al problema real que se estudia, pero afortunadamente para la causa, esto resulta aún más beneficioso, permitiendo inferir según lo reportado una relación en donde una longitud menor y un dominio más pequeño le corresponden un valor de tamaño de ventana menor y por lo tanto menos operaciones por realizarse.

Una vez se conoce el parámetro de ventana del algoritmo DTW, los modelos se encuentran en condiciones para realizar la evaluación final de precisión en la predicción, etapa donde ambos conjuntos de datos se utilizaron en su respectivo contexto: *GesturePebbleZ1_TRAIN* y *GesturePebbleZ1_TEST*. Como se mencionó anteriormente, el conjunto de entrenamiento cuenta con 120 instancias, mientras que la prueba de precisión en predicción se realiza para las 150 instancias del conjunto de prueba. Los datos que se reportan se realizaron fijando los parámetros como se muestra a continuación(Cuadro 2):

Con las cantidades resultantes de las evaluaciones en precisión durante la predicción, se nota una clara ventaja de los modelos basados en características frente al modelo de referencia. Se destaca el éxito del desempeño alcanzado con el nuevo modelo propuesto, siendo notorio en la métrica de precisión al adelantar a SAX-DTW[13] por 3.73 puntos porcentuales; Así como sucede también en el tiempo de predicción, que contrario a las tendencias durante el entrenamiento, se muestra menor en hasta 7.7 segundos o más del doble de rapidez en la predicción.

7. Conclusión y trabajo futuro

Reflexionando sobre los resultados obtenidos en el trabajo, se cumplen las afirmaciones de los artículos en el estado de arte que indican la efectividad y utilidad de algoritmos que propiciaron la simplificación de la complejidad mediante una transformación espacial como SAX, especialmente cuando se comparó con el método de referencia, llegando a ser tan beneficioso que mantiene un valor de tamaño de ventana debajo de 5 y además aporta mayor precisión en

la predicción. El contraste en la complejidad se acentúa con el valor de $w = 16$ cuando se toma en cuenta que la longitud potencial de las series de tiempo que evaluará el algoritmo referencia son de hasta 455 mediciones, mientras que para los métodos SAX-DTW y Atributos SAX-DTW son constantes en 32 y 64, respectivamente.

Se observa para el par de métodos que incluyen la transformación espacial y reducción dimensional de SAX, un aumento de la precisión para la tarea de predicción y una disminución de la complejidad en comparación con el método referencia, por lo que el tiempo de clasificación para obtener la clase de un nuevo gesto también se minimiza. Ayudado de una correcta optimización del algoritmo DTW, las propiedades combinadas de los algoritmos en estos métodos son adecuados para las capacidades de hardware y requerimientos de la propuesta de herramienta embebida.

Por su parte, el modelo de vectores de atributos distancia DTW[22] aún cuando es posible su replicación e implementación experimental, hubiera resultado más costoso que las alternativas evaluadas. Su mayor inconveniente para las restricciones existentes en este proyecto, es la generación del vector de atributos, obligando el cálculo de la similitud DTW para cada una de las instancias de entrenamiento con la serie de tiempo a predecir. Esta metodología podría ser viable cuando se aplica en problemas donde el conjunto de datos de entrenamiento es pequeño y el tiempo de predicción no es crucial.

En términos generales los resultados obtenidos son suficientes para cumplir el objetivo y expectativas esperadas del trabajo con su propuesta Atributos SAX-DTW. El nuevo modelo aprovecha las ventajas que ofrecen cada algoritmo que lo componen y las integra exitosamente bajo el marco metodológico planteado en el trabajo, dotándolo de las capacidades necesarias para superar al estado del arte y presentarse como una alternativa excelente para aplicaciones con restricciones de hardware y rendimiento. Se destaca principalmente su baja complejidad temporal, su disminuido tiempo de cálculo y las modestas exigencias en recursos computacionales, sobre todo para la etapa de predicción.

Retomando las condiciones más generales de la tarea de reconocimiento de gestos, es de notarse que aunque el porcentaje de predicción para ambos modelos es aceptable, e incluso bueno, no se pueden considerar competitivos partiendo desde los resultados reportados en el propio trabajo de Mezari y Maglogiannis[13]. Los modestos resultados obtenidos, se atribuyen a la naturaleza del conjunto de datos con los que se realizó la evaluación, conjunto que incluye únicamente las mediciones de los gestos para el eje espacial Z. El trabajar con series de tiempo multivariables en este tipo de problemas es propicio para conseguir una alta precisión de predicción si los patrones entre ellos son los suficientemente distintos, y esto se deriva de una mayor presencia de rasgos discriminatorios que el clasificador puede aprovechar para definir los límites entre clases.

Con esto en mente se plantea un trabajo a futuro: La evaluación de los 3 modelos bajo los mismos parámetros, o similares, y la misma metodología, utilizando un conjunto de datos propio con instancias de gestos completamente

representadas en los 3 ejes espaciales. El rendimiento obtenido durante este proyecto futuro, permitirá concluir la competitividad del modelo propuesto Atributos SAX-DTW.

El resultado del trabajo a futuro, una vez concluido, se anexará al repositorio público del artículo alojado en la siguiente dirección de GitHub[4].

Referencias

1. National Institute on Deafness and Other Communication Disorders: La afasia. (2017) [En línea]. Disponible en https://www.nidcd.nih.gov/es/espanol/afasia.
2. Instituto Nacional de Trastornos Neurológicos y Accidentes Cerebrovasculares: Apraxia. (2022) [En línea]. Disponible en https://espanol.ninds.nih.gov/es/trastornos/apraxia.
3. American Speech Language Hearing Association: La Disartria. (2021) [En línea]. Disponible en https://www.asha.org/public/speech/Spanish/La-Disartria/.
4. Ismail Fawaz, H., Forestier, G., Weber, J. et al.: Deep learning for time series classification: a review. Data Min Knowl Disc 33, 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1.
5. D. Peña, G. C. Tiao, R. S. Tsay: Introduction. In: A Course in Time Serie Analysis. New York: J. Wiley (2001)
6. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Min Knowl Disc 29, 565–592 (2015). https://doi.org/10.1007/s10618-014-0361-2
7. Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping Chen, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista: Hexagon-ML. The UCR Time Series Classification Archive. (2019) https://www.cs.ucr.edu/ eamonn/time_series_data_2018/
8. A. Bagnall, J. Lines, J. Hills, A. Bostrom: Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles. In: IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 9, pp. 2522–2535 (2015) doi: 10.1109/TKDE.2015.2416723.
9. C. J .G. Ayala Aburto, "Guante traductor de señas para sordomudos,"Tésis título licenciatura, ESIME, unidad Azcapotzalco,. Ciudad de México, México, 2018.
10. E. D. Jiménez Carbajal, G. E. Rivera Taboada, "Sistema de comunicación auditiva para personas con problemas del habla", Tésis para título de licenciatura, ESCOM, Ciudad de México, México, 2013.
11. D. Vishal, H. M. Aishwarya, K. Nishkala, B. T. Royan and T. K. Ramesh, "Sign Language to Speech Conversion,"(en inglés) 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2017, pp. 1-4, doi: 10.1109/ICCIC.2017.8523832.
12. M. M. Chandra, S. Rajkumar and L. S. Kumar, "Sign Languages to Speech Conversion Prototype using the SVM Classifier,"TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 1803-1807, doi: 10.1109/TENCON.2019.8929356.
13. Mezari, Antigoni & Maglogiannis, Ilias. (2017). Gesture recognition using symbolic aggregate approximation and dynamic time warping on motion data. 342-347. 10.1145/3154862.3154927.

[4] https://github.com/LaloValle/SAX-DTW-Features

14. M Müller. "Dynamic Time Warping," *Information Retrieval for Music and Motion*, 4. Alemania: Springer, 2007, pp. 69-73.
15. Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1), 43–49. doi:10.1109/tassp.1978.1163055.
16. Tan, Chang Wei & Herrmann, Matthieu & Forestier, Germain & Webb, Geoffrey & Petitjean, François. (2018). Efficient search of the best warping window for Dynamic Time Warping.
17. Dau, H.A., Silva, D.F., Petitjean, F. et al. Optimizing dynamic time warping's window width for time series data mining applications. Data Min Knowl Disc 32, 1074–1120 (2018). https://doi.org/10.1007/s10618-018-0565-y.
18. Krishnamoorthy, V., 2022. Piecewise Aggregate Approximation. [online] Vigne.sh. Available at: ¡https://vigne.sh/posts/piecewise-aggregate-approx/¿[Accessed 10 May 2022].
19. Krishnamoorthy, V., 2022. Symbolic Aggregate Approximation. [online] Vigne.sh. Available at: ¡https://vigne.sh/posts/symbolic-aggregate-approximation/¿[Accessed 10 May 2022].
20. Petitjean, François & Ketterlin, Alain & Gancarski, Pierre. (2011). A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognition. 44. 678-. 10.1016/j.patcog.2010.09.013.
21. Minnaar, A., 2022. Time Series Classification and Clustering with Python –. [online] AlexMinnar. Available at: ¡http://alexminnaar.com/2014/04/16/Time-Series- Classification-and-Clustering -with-Python.html¿[Accessed 10 May 2022].
22. Kate, R.J. Using dynamic time warping distances as features for improved time series classification. Data Min Knowl Disc 30, 283–312 (2016). https://doi.org/10.1007/s10618-015-0418-x