Revisión de las técnicas existentes para la clasificación de series de tiempo en el contexto de un sistema traductor de gestos motrices a habla

Reporte Técnico de un trabajo de investigación

Proyecto Intermedio

*Alumno: *Valle Martínez Luis Eduardo*

Profesor: Dr. Ricardo Barrón Fernández

**e-mail: lvalle212@alumno.ipn.mx*

Palabras clave - Algoritmo *Dynamic Time Warping*(DTW), Clasificación de Series de Tiempo, Modelo *Symbolic Aggregate Approximation*(SAX)

1. Introducción

El lenguaje es la herramienta y habilidad humana más importante que hemos sido capaces de desarrollar como especie, al punto que su invención y dominio ha sido el factor primordial que identifica nuestro intelecto y ha sido indudablemente la pieda angular de la que nos hemos valido para alcanzar el nivel de desarrollo del que gozamos hoy en día en el mundo moderno.

A nivel oral los humanos expresamos el lenguaje mediante el habla, y que es naturalmente el medio principal universal por el que nos comunicamos con los demás. Precisamente por su importancia en el uso cotidiano en la vida diaria de las personas, la existencia de afectaciones que impidan o imposibiliten la expresión oral presenta un panorama complejo para desarrollarse como normalmente cualquier otra persona podría.

Pacientes con afecciones del habla como las afasias, apraxias y disartrias, se enfrentan a un problema neurológico ocasionado en su mayoría por accidentes cerebrovasculares, contusiones, o daños en el lóbulo frontal izquierdo del cerebro, relacionado a las funciones lingüisticas, lo que de alguna u otra forma, les afecta en su expresión de forma motriz o neurológicamente.

Frente a esta problematica se propone una primera aproximación de solución como proyecto de Trabajo Terminal, y que consiste en la creación de un sistema embebido y prototipo de guante *wearable* sensor de los movimientos con una mano a través del uso del SoC micro:bit, del que principalmente se realiza el muestreo de los cambios en la aceleración en los 3 ejes espaciales, permitiendo la identificación de gestos correspondientes a movimientos de un código motriz también propuesto al usuarse un modelo de *Machine Learning* para la clasificación de los datos recopilados y la identificación de una clase textual con la cual se le permite al usuario conformar palabras. En una etapa siguiente y final, se lleva a cabo la reproducción sonora de las palabras haciendo uso de servicios de traducción de texto a habla.

En este sentido, este trabajo se enfoca en la investigación de los modelos y técnicas existentes en el *Machine Learning* para la clasificación de los datos en series de tiempo, pudiendo de esta forma aportar directamente al desarrollo de la solución del TT, al estudiarse las posibilidades de modelos, sus características e incluso proponiendo una técnica propia basándose en estudios previos, que con documentación en la implementación experimental han logrado probar su precisión y buen desempeño para este tipo de tareas.

Abarcando una gran cantidad de estudios realizados, se encuentra el algoritmo DTW, el cual es un recurso utilizado popularmente por décadas para la medir la similitud entre series de tiempo sin importar la velocidad entre estas, o el desafase, realizando para esto una deformación de la serie de tiempo en búsqueda de una alineación óptima entre ambas señales.

Ya dentro del ámbito de la tarea de la clasificación, por décadas se ha posicionado como un modelo estándar de referencia una técnica combinada que implementa DTW como función de distancia para el agrupamiento no

supervisado realizado por el modelo de *clustering* 1-NN, y aunque se trata de probablemente la solución más sencilla de entre los modelos existentes en *Machine Learning*, se ha mantenido imbatible en precisión por una gran mayoría de modelos mucho más complejos.

Se explica el algoritmo DTW, así como los esfuerzos que se han realizado en la aplicación de restricciones que han permitido disminuir la complejidad del proceso, de esta forma optimizando su cálculo. Así también se muestran soluciones de un par de trabajos más contemporáneos que aportan soluciones de modelos candidatos para remplazar a la técnica estándar DTW 1-NN, mostrando su considerable mejora frente al desempeño que muestra el modelo referencial.

Derivada de la revisión y estudio de ambas técnicas, se realiza la proposición teórica de un modelo clasificador de secuencias temporales con perspectiva a la solución del Trabajo Terminal. Tomando las características más valiosas que hacen destacar a cada uno de los 2 modelos y utilizando en combinación el algoritmo DTW y el modelo SAX, se tiene como objetivo con esta propuesta poder adoptar de forma general en 1 sola técnica todas estas bondades que permitan desarrollar la tarea con gran precisión y eficacia.

2. Tópico a desarrollar o problema a resolver

Apectos como los gestos, muecas, sonidos, etc. son factores influyentes en la comunicación del estado consciente o inconsciente de una persona que lo hace visible para ser interpretable por los demás. Sin embargo en la comunicación humana, aunque relevantes estos aspectos, es inevitable reconocer al lenguaje como el elemento simbólico fundamental en la interacción entre individuos y grupos.

El lenguaje humano como la prueba de especímenes que han desarrollado un intelecto sofisticado, ha sido en la historia de la evolución humana la piedra angular de la que nuestra especie se ha valido como herramienta para generar conocimiento y perpetuarlo para ser compartido y adquirido por generaciones posteriores, que ha facilitado la comunicación para la organización de grupos con un objetivo en común, y que incluso ha fungido como el medio para la expresión de sentimientos y pensamientos en forma de literatura y poemas.

A nivel oral, expresamos el lenguaje mediante el habla, alcanzado en algún punto de la evolución del sistema canal vocal-auditivo en el humano, derivado del descenso de la laringe y lo que nos dio la posibilidad de crear sonidos[1]. A través de la especialización como especie en la actividad del habla, se fueron otorgando semántica a los sonidos generados y nos permitió la asociación de un significado a estos[1].

Tratándose de la herramienta cotidiana que permite la comunicación e interacción social con los demás, es natural la relevancia que se le otorga al lenguaje en la vida del humano, resaltando específicamente su derivación oral como uno de los tema de interes en el estudio de este trabajo. La vivencia ordinaria diaria se dislumbra compleja cuando se restringe el uso del lenguaje verbal, y sin embargo la realidad presenta la existencia de sectores de la sociedad que a raíz de trastornos y afecciones, afrontan dificultades que parcial o completamente, les impiden expresarse oralmente.

A continuación se mencionan las afectaciones más habituales que tienen la posibilidad de beneficiarse de un prototipo de solución que implementa los resultados de este documento. Las primeras son las **afasias**[2-4], las cuales son problemas médicos originados por una lesión cerebral y que resulta en la pérdida o alteración del lenguaje. Las **apraxias**[2,5,6] que son trastornos neurológicos caracterizado por la pérdida de la capacidad de llevar a cabo movimientos diestros y gestos, aún cuando se tenga el deseo y la habilidad física para hacerlo, teniendo diferentes afectaciones en función de la parte lesionada en el cerebro. Finalmente existe la **disartria**[2,7,8] que es un trastorno de la ejecución motora del habla debido a un problema neurológico por la presencia de un accidente u lesiones cerebrales. Afecta gravemente la motricidad de los músculos para el habla. Importante mencionar que en algunos casos los profesionales médicos recomiendan el uso de algún dispositivo electrónico o tecnológico de apoyo para la comunicación para las disartrias.[8]

Otro tipo de trastorno que imposibilita de forma indirecta la capacidad para la expresión hablada, aunque como consecuencia de la afección principal, es la sordera de percepción total y es clasificada como un trastorno de la audición. Este proyecto no considera a este sector de la población como el público objetivo principal por razones

como: El amplio desarrollo del Lenguaje de Señas como principal recurso de comunicación, y las limitantes propias del trabajo donde no se provee una solución para una comunicación bidireccional con usuarios con este tipo de afección.

En este sentido y debido a la aparente utilidad e innovación en el uso de guantes de traducción para la comunicación entre personas con afectaciones en el habla y audición con la sociedad en general, se han desarrollado cantidad de trabajos a escala internacional, nacional e incluso interinstitucional basándose principalmente en el lenguaje de señas respectivo del país donde se investigó.

Las mayoría de las propuestas de solución en los trabajos consultados[9,10,11,12] tienen por tendencia el uso extendido de prototipos guantes, o incluso *wearables*, con los que se realiza la detección para la interpretación de movimientos y gestos con las manos, sin embargo esta clase de solución cuenta con inconvenientes principalmente físicos, como la cantidad de hardware requerida, la portabilidad del prototipo y la falta de estética para un uso cotidiano generalizado. En otra rama tecnológica pero con soluciones existentes en cantidades casi iguales, se tienen los proyectos enfocados en soluciones basadas en Visión Artificial, las cuales suelen ser mucho más cómodas en términos de portabilidad para el usuario, pero en constraste requieren de un ambiente de iluminación y constraste controlados para adquirir resultados aceptables con las técnicas de análisis de imágenes.

Tomando en cuenta estas consideraciones, se propone una solución que será desarrollada como Trabajo Terminal para la titulación de la Ingeniería en Sistemas Computacionales, y de la que toma provecho el trabajo de investigación plasmado en este documento para más tarde una implementación práctica. Esta propuesta se decanta por un prototipo físico *wearable* tipo guante, con el que se busca aminorar el impacto de portabilidad y estética al utilizarse un único dispositivo sensor, el SoC micro:bit, que integra un IMU del que principalmente se utiliza para el muestreo de los patrones de movimiento el acelerómetro. Estos patrones manuales serán elementos pertenecientes a un código motriz especialmente propuesto para este sistema, y con los que se busca clasificar cada patrón en una de las clases textuales con las que se logra conformar el texto, a través de la implementación de un modelo clasificador de *Machine Learning*.
Una última etapa, ejecutada después de la conformación de las palabras o frases, se logrará mediante la consulta de un servicio de *Text-to-Speech* para la reproducción sonora del texto armado. El proyecto *TTS de Mozilla*, es una solución que se ejecuta como servicio en red local, aunque también se considera el uso de servicios especializados en la nube, como por ejemplo *Microsoft Azure Text-to-speech*.

El aporte que se consigue de este trabajo de investigación se encuentra en el informe de las diferentes alternativas de modelos y técnicas contemporáneas para la tarea de clasificación, así como sus características, limitaciones y beneficios, todo esto en vista de la adopción posterior de un modelo basado en *Machine Learning* que permita de una manera eficiente y con resultados aceptables, la clasificación de los patrones motrices muestreados por el prototipo *wearable* sensor que se porta en la mano, permitiendo identificar la clase textual del patrón que permita realizar su conversión a texto. Al final este proceso se describe como traducción de Movimiento a Texto.

Estos patrones de movimiento que se clasificarán son resultado de muestrear la fuerza de aceleración en los 3 ejes espaciales durante la realizacíón de un movimiento definido en el código motriz propuesto. Este muestreo provee en la salida una serie de tiempo con 3 características, o en su caso 3 series de tiempo con una variable espacial objetivo.

3. Estado del arte

La clasificación de serie de tiempo es una tema de gran importancia en diferentes ambientes y disciplinas, principalmente útil para la toma de decisiones. A medida que los años han pasado, gran esfuerzo se ha dedica en la investigación de nuevos y mejores modelos que resulten tan precisos como sea posible para la clasificación, siendo importante mencionar el desempeño destacado que se observa en un modelo sumamente sencillo que consiste en la conjunción del algoritmo DTW, como función de medida de la similitud entre secuencias temporales, con el algoritmo de agrupamiento *k-NN*[13]. Se trata de uno de los algoritmos más veteranos en el tema y sin embargo dificilmente se cuenta actualmente con modelos que puedan superar contundentemente su precisión, valiéndole entre la comunidad la cualidad del modelo referencia base predilecto para la comparación de precisión y eficiencia entre demás modelos clasificadores de series secuenciales.

Aún con esto, los estudios en el tema siguen desarrollandose y se encuentran nuevas alternativas que también aprovechan las características de la medida DTW en la similitud entre series, pero utilizando otros modelos más complejos de *Machine Learning* obteniendo una mejora significativa al par histórico de DTW y el clustering 1-NN. En el trabajo [14] se propone el uso de DTW no solo como función de medida para el algoritmo de agrupamiento, sino que sus magnitudes resultantes conforman el vector de características que mas tarde es usado para la clasificación con cualquier método estándar de *Machine Learning*, lográndose la conformación del patrón en donde cada elemento será el resultado de la medida del camino óptimo de DTW cuando se compara la serie de tiempo en cuestión con cada una de las series de tiempo pertenecientes al conjunto de entrenamiento. Finalmente se utiliza un modelo SVM por su grán precisión aún cuando se cuenta con vectores de un gran número de características, pero se insinua en el *paper* que utilizando esta técnica cualquier otro modelo clásico de clasificación puede usarse en vez.

Finalmente en el trabajo [14] se muestra una mejora en el proceso de clasificación cuando se utiliza en conjunto la técnica de conformación de características usando DTW, y el modelo SAX para la representación de características de una serie de tiempo en un espacio dimensional menor de 'palabras' simbólicas. Este acercamiento únicamente propone la combinación de ambos vectores de caracteristicas y que funcionará para alimentar el modelo, el cuál también corresponde a un modelo máquinas de soporte vectorial en el trabajo, pero puede utilizarse cualquier modelo clasificador.

El *paper* [15] describe un trabajo sumamente relacionado a la solución propuesta para el Trabajo Terminal, consistiendo en un sistema capaz de reconocer 6 gestos motrices utilizando un *smartwach* como dispositivo sensor de la aceleración en los 3 ejes espaciales. Para la clasificación de los patrones se proponen y evalúan 3 técnicas, todas ellas utilizando el modelo de agrupamiento 1-NN para la clasificación.

El primer par de técnicas consiste en calcular la distancia entre 2 series de tiempo utilizando DTW con una medida de costo local L1(también nombrada *Cityblock Distance*) y SAX, obteniéndose un porcentaje de precisión del 93.15 % y 96.44 % respectivamente.

Finalmente se evalua una tercer técnica que combina la representación de la serie de tiempo en una cadena alfabética del método SAX, y la distancia entre símbolos se realiza encontrado la alineación óptima del par de cadenas alfabéticas con DTW. Esta técnica de las 3 es la que mayor precisión obtuvo con un 99.21 % y que combina las mejores características de ambos métodos, la efectividad y baja complejidad de SAX con la insensibilidad a la fluctuación de la velocidad durante la execución del movimiento de DTW.

Cada vez se encuentra a disposición de la sociedad general una mayor capacidad de poder computacional, y esto se ha visto claramente con el alza de adquisición y uso de las Unidades de Procesamiento Gráfico(GPUs), lo que inevitablemente ha permitido extender la influencia de modelos que aprovechan la alta capacidad de paralelización que ofrece este hardware, especialmente los modelos de *Deep Learning*.

Siempre en busca de alternativas de aumentar el desempeño de la tarea de clasificación con el modelo DTW 1-NN como referencia, se encontró que la técnica de ensamblado(*ensembling*) de modelos superaba la implementación de un modelo individual, por lo que se volvieron populares en muchos trabajos, siendo especialmente populares los ensambles de árboles de decisión(Random Forests), ensambles de diferentes tipos de clasificadores discriminadores(SVM) en una o varios espacios de características.

Aún cuando estos ensambles superan al modelo base, se destacan estas teécnicas por tener en común la transformación de los datos a series de tiempo en un nuevo espacio de características, motivando precisamente el desarollo de un popular ensamblado llamado COTE(*Collective Of Transformation-based Ensembles*), que no solo incluye ensamblados de diferentes clasificadores sobre la misma transformación, en cambio ensambla diferentes clasificadores sobre diferentes representaciones de series de tiempo[16].

Tiempo después se realizó una extensión con un sistema jerárquico de votación, HIVE-COTE, el cuál ha mostrado una mejor significativa apalancándose de una nueva estructura jerárquica con votación probabilística, incluyendo 2 clasificadores nuevos y 2 repressentaciones adicionales del dominio de transformación, siendo actualmente el estado del arte de la clasificación de series de tiempo[16].

Este método de ensambles cuenta con un gran detrimento, su inmensa intensidad computacional e impracticidad para problemas de big data o aplicaciones de tiempo real, pues su método requiere el entrenamiento de 37 clasificadores, como así también la validación cruzada de cada hyperparámetro de los algoritmos, sin mencionar que no solo la etapa de entrenamiento pero también la de clasificación toman un gran tiempo computacional para finalizar.

Frente a este panorama algunos trabajos han explorado la factibilidad de modelos alternativos basados en *Deep Learning* para exitosamente conseguir la tarea de clasficación en series temporales, tales como modelos CNNs(*Convolutional Neural Networks*), DNN(*Deep Neural Networks*), FCN(*Fully Convolutional Neural Networks*), ResNet(*Residual Networks*), etc[16].

4. Fundamentos y Herramientas

Series de Tiempo

Una **Serie de tiempo** es una colección de observaciones realizadas secuencialmente en intervalos regulares de tiempo[17,18], siendo una característica especial de estas el hecho de que estas observaciones no son independientes y que su análisis debe tomar en cuenta el orden en el tiempo de las observaciones[13]. Muchas áreas de estudio recolectan datos en forma de series de tiempo tales como negocios, economía, ingeniería, medicina, ambiente, etc.[18]

Se definen 2 tipos de serie de tiempo en función del muestreo de sus observaciones, de forma que cuando las observaciones se realizan continuamente en el tiempo estas series de tiempo son nombradas **continuas**. De la misma forma si una serie es medida con observaciones tomadas unicamente en tiempos específicos, usualmente espaciados igualmente, se nombran **discretas**, y este término sigue siendo válido incluso cuando la variable en ser medida es continua[17].

Generalmente el manejo de series de tiempo en un contexto práctico se estará hablando de series discretas. Esto se debe a que la medición de cualquier variable, ya sea física o dentro de un sistema lógico, se realiza mediante el muestreo del evento con una tecnología digital, implicando la medición de su valor en un instante de tiempo siendo imposible para cualquier tipo de sensor el muestreo continuo por el proceso propio de la digitalización.

Finalmente referente a las series de tiempo cuando las mediciones son dependientes, entonces los valores futuros pueden ser predecidos de observaciones pasadas, incluso dándoseles el nombre de series **determinísticas** cuando las pedicciones son exactas utilizando observaciones anteriores. Por su parte cuando el futuro de una serie de tiempo esta parcialmente determinado por sus valores pasados, las predicciones exactas son imposibles de realizar, requiriendo remplazar entonces la idea a que los futuros valores tienen una distribución de probabilidad que está condicionada por el conocimiento previo, a este tipo de series se les llaman **estocásticas**[17].

DTW[19]

Dynamic Time Warping(DTW) es una técnica bien conocida para encontrar una alineación óptima entre 2 sequencias dependientes del tiempo(series de tiempo) dadas y bajo ciertas restricciones[13]. Este algoritmo es sumamente util para medir la similaridad entre dos secuencias temporales que no se alinean exactamente en el tiempo, velocidad o extensión[13]. Originalmente este algoritmo había sido utilizado para comparar patrones de habla en reconocimiento de habla automático, siendo aplicado en otros campos de forma exitosa para lidiar con deformaciones en el teimpo y diferentes velocidades.

DTW Clásico

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.

$$X = (x_1, x_2, ..., x_N) \quad N \in \mathbb{N}$$
$$Y = (x_1, x_2, ..., x_M) \quad M \in \mathbb{N}$$

Se fija un *espacio de características* denotado por $\mathcal{F}$

$$x_n, y_m \in \mathcal{F}$$
$$\text{para } n \in [1:N] \text{ y } m \in [1:M]$$

La comparación de dos características diferentes se realiza mediante una *medida de costo local*, también llamada *medida de distancia local* definida por:

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

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

Tipicamente *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.

Formalmente un *warping path* 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 *(N,M)-warping path* $p = (p_1, ..., p_L)$ 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 (1,*a)*, donde utilizando la matriz de costo acumulado se marca una secuencia en color verde de las distancias óptimas satisfaciendo siempre las condiciones(i,ii,iii).

Con la condición *(i)* se asegura que el primer elemento tanto de X como de Y, y el último elemento de X y el último de Y se encuentren alineados.

La condición de monotonicidad refleja el requisito de una correspondencia temporal, así si un elemento en X precede a un segundo, esto también debería ser válido para los elementos correspondientes en Y.

Finalmente la condición *(iii)* expresa un tipo de continuidad, de esta forma no hay elemento en X y Y que pueda ser omitido y no hay réplicas en la alineación.

El costo total será:

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

Escogida una distancia específica el costo total cambiará su forma de ser calculado. Por ejemplo para una distancia Euclideana que es una distancia muy típica:

$$c_p(X, Y) = \sqrt{\sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell})} = \sqrt{\sum_{\ell=1}^{L} (x_{n_\ell} - y_{m_\ell})^2}$$

Y para la distancia L1, *Manhattan* o *City Block*:

$$c_p(X, Y) = \frac{1}{L} \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell}) = \frac{1}{L} \sum_{\ell=1}^{L} |x_{n_\ell} - y_{m_\ell}|$$

Y debido a que el camino óptimo entre ambas series es el camino con $p*$ teniendo el costo total mínimo entre todos los posibles caminos, la *distancia DTW(X,Y)* se define como el costo total de $p*$.

$$\begin{aligned} DTW(X, Y) &= c_{p^*}(X, Y) \\ &= min\{c_p(X, Y) | p \text{ es un } (N, M) - \text{warping path}\} \end{aligned}$$

Su determinación puede conllevar probar cada camino entre X y Y, sin embargo este procedimiento llevaría a una complejidad computacional exponencial en las longitudes *N,M*. En este punto se introduce una complejidad *O(NM)* utilizando un algoritmo basado en *programación dinámica*, y con la que se requiere la definición de la *matriz de costo acumulada D(n,m)* utilizando la función de costo *DTW(X,Y)*:

$$D(n, m) = DTW((x_1, ..., x_n) = X(1 : N), (y_1, ..., y_m) = Y(1 : M))$$

Finalmente construida la *matriz de costo acumulado D*, se puede obtener el camino óptimo único entre el par de series de tiempo y su costo total realizando la suma de las células:

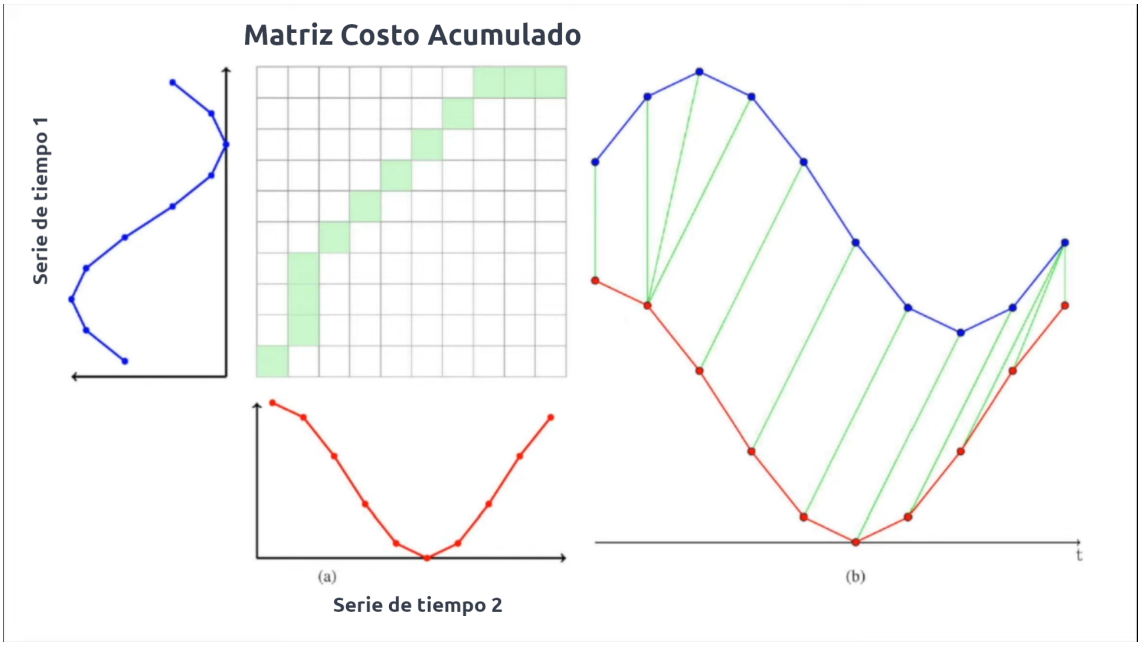

Figura 1: a) Matriz de costo acumulada del cálculo de la distancia entre la Serie de tiempo 1 y Serie de tiempo 2. Los recuadros verdes conforman el camino de deformación óptimo(*optimal warping path*) p^*.
b) Resultado de la alineación óptima para la Serie de tiempo 1 en la Serie de tiempo 2 utilizando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

Algoritmo: *OPTIMAL WARPING PATH*

Entradas: Matriz de costo acumulada D
Salidas: *Warping Path* óptimo $p*$

$$p_{\ell-1} = \begin{cases} (1, m-1) & \text{if } n = 1 \\ (n-1, 1) & \text{if } m = 1 \\ argmin\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} & \text{de otra forma} \end{cases}$$

Un ejemplo del resultado obtenido de este algoritmo se observa en la Figura (1,*b)*.

Restricciones en DTW

Una variante comun de DTW es la imposición de condiciones restringidas globales on el camino *warping* admisible. Este tipo de restricciones tienen 2 beneficios:

1. Prevenir alineaciones patológicas reduciendo de esta forma incrementando la precisión del clasificador. Se evita que el camino óptimo posible se desvié demasiado de la diagonal principal.

2. Reducir el tiempo de complejidad de DTW de $O(L^2)$ a $O(w \times L)$ con $0 \geq w \leq L-1$, donde w define una banda donde se calcula el camino óptimo como un subconjunto del espacio global. Cuando $w = 0$ corresponde a la distancia Euclideana y con $w = L-1$ corresponde al DTW clásico sin restricciones.

Se tienen 3 métodos principales para la definición de esta banda: banda Sakoe-Chiba[20] también llamada *warping window*, Paralelograma Itakura y banda Ratanamahatana-Keogh. En este estudio se considera únicamente la banda Sakoe-Chiba o ventana de deformación, pues se trata del método más popular y de uso más extendido en la literatura.

Banda Sakoe-Chiba

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 **??**.

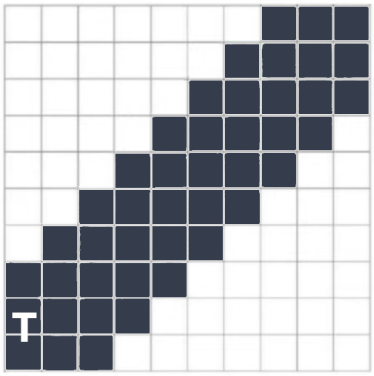

Figura 2: Banda de Sakoe-Chiba que corre sobre la diagonal principal de la matriz y que tiene un ancho fijo T

La implementación de esta restricción incrementa considerablemente la velocidad de cálculo computacional de DTW, disminuyendo la complejidad algorítmica a $O(T \times max(N, M))$ con típicamente $T \ll M$ y $T \gg N$. Además de la disminución en la complejidad, es conocido la fuerte influencia que tiene el recorte del espacio de cálculo mediante una ventana en la precisión conseguida, permitiendo incluso a través de una ventana con tamaño aprendido, reducir el porcentaje de error desde un 35 % a un 7 %.

A través de los años se ha logrado confirmar mediante la extensa conducción de experimentos que el tamaño de la WW debe ser obtenido via el cálculo de una validación cruzada, asegurando que esta medida se competitiva para el actual conjunto de entrenamiento[21], siendo crítico el tamaño de esta para la optimización del error de predicción o precisión.

Normalización de series de tiempo[22]

La normalización de los datos en una serie de tiempo es un proceso realizado con el objetivo de obtener un mejor desempeño de algoritmos en *Machine Learning* si la serie de tiempo tiene una escala consistente o distribución, existiendo un par de técnicas que permiten consistentemente rescalar una serie de tiempo: **Normalización** y **Estandarización**.

La normalización consiste en reescalar los datos de un rango original de forma que todos los valores se encuentren dentro del rango 0 y 1. Este proceso suele ser útil e incluso requerido un algunos algoritmo de *Machine Learning* cuando se cuenta con series de tiempo con valores de ingreso que difieren en escalas. Es aprovechada esta transformación por algoritmos como k-NN, Regresión Lineal y RNA que ponderan los valores de entrada.

$$y = \frac{x - min}{max - min}$$

Figura 3: Ecuación normalización

Estandarización o normalización-z, implica el reescalado de una distribución de valores de forma que la media de los valores observados sea 0 y la desviación estándar 1. Esto consiste en ajustar los datos en una distribución Gausiana(curva de campana) con una media bien definida y desviación estándar, siendo común asumir que estas

observaciones respestan esta distribución, siendo sin embargo posible realizar la estandarización cuando los datos no cumplen esta expectativa.

Algoritmos que se benefician de la estandarización para un mejor desempeño son Máquinas de Soporte Vectorial(SVM), Regresión Lineal y logística, entre otros modelos.

Este proceso requiere el conocimiento de una estimación precisa de la media y desviación estándar de los valores observables:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{\sum (x-\mu)^2}{n}}$$

Figura 4: Ecuación de la media y desviación estándar

Piecewise Aggregate Approximation(PAA)[23]

Piecewise Aggregate Approximation(PPA) es un algoritmo con la idea básica de 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. 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:

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

El aspecto más interesante del algoritmo es como se crean los segmento de mismo tamaño. Es importante notar que antes de realizar la aproximación promedio de cada ventana, el vector debe ser z-normalizado, y una vez haya sido estandarizado el vector la aproximación por partes se calcula.

Se identifican 2 casos escencialmente durante la separación de segmentos. Cuando $m < n$ y m es múltiplo de n por lo que el residuo de su división es 0. Este caso indica que la división del vector se puede realizar facilmente en ventanas de igual tamaño, siendo directo el cálculo del promedio para cada segmento.

Sin embargo cuando se tiene el caso donde $m < n$ y m no es múltiplo de n, deja de existir un balance que evita la división exacta del vector en segmentos de mismo tamaño. Lo que ahora se requiere es que cada ventana sea redimensionada de forma que cada elemento de la serie de salida es el promedio de un segmento de mismo tamañao al vector de entrada.

Symbolic Aggregate Approximation(SAX)[15,24]

SAX es una técnica desarrollada para reducir la dimensionalidad 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 $A = a_1, a_2, ..., a_3$.

La primer parte de la discretización es manejada por la transformación PAA. El siguiente proceso es la asignación de símbolos para cada sección, siendo requerimiento para esta segunda parte de discretización es que los símbolos sean asignados equiprobablemente(propiedad de una colección de eventos teniendo la misma probabilidad de ocurrir). Esto se cumple facilmente por la previa normalización de los datos de la serie en una distribución Gausiana, por lo 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 proceso del algoritmo toma los siguientes procesos en el siguiente orden[15]:

1. Estandarización o normalización-z de los datos de la serie de tiempo para tener una media de 0 y una desviación estándar de 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.

Puntos de quiebre[24]

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 $\beta_{i+1} = \frac{1}{\alpha}$

Estos puntos de quiebre pueden ser obtenidos mirando una tabla estadística y puede ser utilizada para 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 ínidice, y así se sigue.

Palabras

Una *Palabra* es una secuencia C de longitud c que puede representarse como una palabra. Si a_i es el i-ésimo elemento del alfabeto, entonces el mapeo de aproximación de PAA a la palabra $\widehat{C}$ es obtenida usando $\widehat{C} = a_j$ si $\beta_{j-1} \leq \overline{C}_i < \beta_j$. Esto define la representación simbólica de la serie.

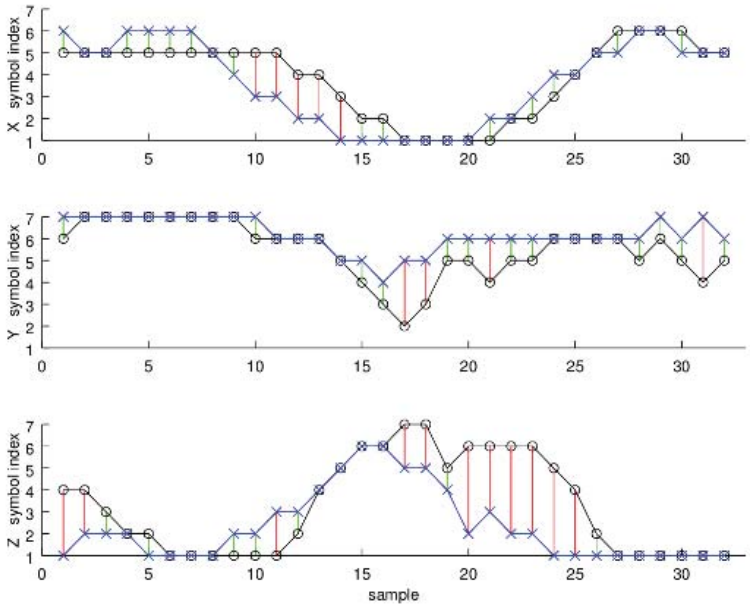

Figura 5: Ejemplo de la representación producida por las Palabras SAX para 2 series de tiempo que corresponden a un gesto realizado por personas diferentes.
Recuperado de *The Sax method*[Figura 4] de Gesture recognition using Symbolic Aggregate approximation and Dynamic Time Warping on Motion Data.

5. Plan de solución o exposición del tópico

El objetivo de este trabajo es recopilar los modelos utilizados típicamente para la clasificación de series de tiempo, teniendo particular atención en trabajos relacionados con la clasificación de patrones de movimiento mediante el muestreo de valores en aceleración u similares; Describir los recursos y funcionamiento de los modelos existentes para *Machine Learning* y avances en propuestas de *Deep Learning*. Finalmente planteando una técnica para la clasificación de patrones motrices que busca conseguir los beneficios de modelos existentes sin tratarse de un modelo completamente nuevo pero más bien una nueva variación de la técnica.

En una primera instancia se inicia con el estudio de las técnicas existentes en el ámbito del *Machine Learning* para la clasificación de series de tiempo, siendo estos modelos los más populares y más ampliamente utilizados en proyectos de incluso décadas de antigüedad. Se tiene una gran ventaja con este tipo de modelos, pues su extenso uso acredita los beneficios al implementarse en cantidad de proyectos, haciendo evidente también las posibilidades de mejora u optimización, influyendo en la creación de técnicas alternativas que buscan mejorar 1 o varios aspectos de estos robustos modelos.

En una segunda instancia se investigan las soluciones aportadas por técnicas basadas en *Deep Learning*, siendo en su mayoría modelos noveles, o al menos no de un uso tan extendido como aquellas de *Machine Learning*. Aunque cada vez existen más propuestas dentro de este ámbito, el uso del *Deep Learning* para la clasificación de series de tiempo es reducido con respecto a técnicas más sencillas, por lo que se estudian algunas propuestas con sus beneficios y restricciones en esta área para cumplir con el objetivo principal del trabajo.

Un buen punto por donde partir es a través de una combinación del algoritmo DTW y el algoritmo de agrupamiento no supervisado *k-NN*(empíricamente se han identificado mejores resultados con 1-NN), para la clasificación de sequencias dependientes del tiempo pues ha mostrado ser un clasificador sumamente preciso y robusto en comparación con técnicas más poderosas[25].

Esta aproximación para clasificación de series de tiempo es probablemente la solución más sencilla por aplicar, y sin embargo aún con las décadas que han pasado desde su primera implementación, las numerosas medidas de distancia desarolladas, representaciones de características, y metodos especializados de *Machine Learning* propuestos para la clasificación de series de tiempo, dificilmente se ha podido vencer contundentemente esta sencilla combinación.

Partiendo de su robustez y sencilla implementación, el mayor esfuerzo se centra en optimizar el algoritmo DTW con diferentes consideraciones, disminuyendo así el tiempo de ejecución y procesamiento computacional que mayoritariamente se ocupa para el cálculo de la tabla dinámica *matriz de costo acumulado* con un tamaño *NxM* y complejidad *O(NM)*.

Se describen las técnicas creadas y expuestas en los papers[14][15], que principalmente utilizan el algoritmo DTW por todas las posibilidades que este ofrece en este problema de clasificación. Finalmente en lo que refiere a *Machine Learning*, tomando como referencia este par de trabajos se realiza una propuesta de modelo que tendrá como objetivo trasladar a una misma técnica las características ventajosas de cada modelo.

Importante resaltar que esta última técnica propuesta para la clasificación de series de tiempo es planteada teóricamente tomando en consideración los algoritmos, técnicas y observaciones realizadas en el par de trabajos previos que aseguran según sus resultados, conseguir con estos una mejora significativa en la precisión con respecto al modelo base DTW 1-NN. La verificación práctica experimental de su eficacia frente a los modelos estudiados abre la posibilidad a un trabajo posterior que cumpla con este objetivo de mostrar resultados de implementación, sobrepasando su realización en este trabajo los alcances de la investigación.

Después de revisar las técnicas existentes en el *Machine Learning*, se describen las alternativas de aproximación a este problema con modelos de *Deep Learning*, habiendo propuestas con mayor popularidad y de amplio uso en proyectos en tiempos más contemporaneos, dejándose atrás técnicas de mayor trabajo manual y procesamiento del vector de características para tomar directamente los valores de la serie temporal a las entradas del modelo *Deep Learning*.

6. Desarrollo o solución

El problema de la clasificación de series de tiempo involucra el entrenamiento de un clasificador en un conjunto de casos, donde cada caso contiene un conjunto ordenado de atributos en valores reales y una etiqueta de clase[26]. Este problema, y cada vez más influenciado por el poder computacional disponible que permite la obtención de los datos en secuencias dependientes del tiempo, tiene aplicaciones en varios dominios como el médico, biologico, financiero, *data mining*, estadísticas, procesamiento de señales, procesamiento de imágenes, etc.

A partir del enorme interes desencadenado por esta tarea, investigadores han dedicado un esfuerzo considerable proponiendo cantidad de métodos en las décadas pasadas que cáen en 2 tipos de categorias[14]:

- Métodos basados en **distancia**: Una función de distancia es definida para calcular la similaridad entre 2 series de tiempo, así cuando a una secuencia es tipicamente clasificada como perteneciente a la misma clase como su serie de tiempo más cercana presente en los datos de entrenamiento de acuerdo a la función de distancia. Distancia Euclideana, DTW, *Longest Common Subsequence*, etc.
- Métodos basados en **características**: Una representación estadística o simbólica es primeramente definida para la serie de tiempo, y entonces se emplea uno de los tantos métodos de *Machine Learning* entrenados para la clasificación de series de tiempo utilizando el conjunto de entrenamiento que utiliza la representación basada en características.

Se ha encontrado que la forma más sencilla de obtener una mejora en la precisión de la clasificación consiste en un método basado en características, donde se realiza una transformación de los datos en un espacio alternativo en el que las características descriminatorias pueden ser más fácilmente detectables que el diseñar un clasificador más complejo en el dominio del tiempo[26], y sin embargo a pesar de esta consideración, las numerosas medidas de distancia utilizadas, otro tipo de representaciones en características y métodos especializados de *Machine Learning* en la tarea de clasificación de series de tiempo, la técnica de décadas de antigüedad que implementa en combinación DTW y 1-NN(*Nearest Neighbour*), no ha sido posible de vencer contundentemente en precisión.

A pesar de lo que se esperaría como sucede en otras tareas de clasificación donde un método de *Machine Learning* más poderoso pudiera significativamente superar el rendimiento de DTW 1-NN, pues es quizás el método de *Machine Learning* más sencillo, no ha sido este el caso para la clasificación de series de tiempo, valiendole con el tiempo la aceptación entre la comunidad como el estándar dorado contra el que comparar medidas alternativas en esta tarea.

DTW 1-NN

Esta robusta técnica combina la medida de similitud entre secuencias de tiempo DTW y el algoritmo de agrupamiento no supervisado k-NN(*k Nearest Neighbours*) en su modelo con *k=1*, parámetro definido de forma empírica por inifindad de trabajos anteriores y que actualmente conforma la variante de este algoritmo más preciso y utilizado por la comunidad.

En esta variante del algoritmo k-NN, en vez de implementarse la típica distancia Euclideana para el cálculo de la posición de los centroides, los baricentros se calculan con respecto a DTW utilizando el algoritmo DTW *Barycenter Averaging*(DBA)[27], que se encarga de minimizar la suma de distancias cuadradas DTW entre un conjunto de series de tiempo; Y se utiliza como métrica DTW también para definir la distancia entre series con el centroide para su agrupamiento[13].

A partir de su extendida aplicación en diferentes implementaciones, se ha identificado su gran capacidad de clasificación con una gran precisión, y más sin embargo se aprecia la sencillez que ofrece esta técnica en detrimento de la complejidad de tiempo en cálculo principalmente por el algoritmo DTW, que tiene una complejidad $O(N^2)$ cuando se evalua la similitud de series de tiempo con igual número de elementos en extensión, variando en función de la extensión de ambas series de tiempo.

A partir de esto, se ha considerado la optimización del algoritmo utilizando restricciones que permita la disminución de la complejidad en la conformación de la matriz de costo acumulada, siendo el estándar un tamaño de ventana de deformación a través de validación cruzada.

Optimización de la ventana de deformación(*warping window*)[28]

El aprendizaje de la ventana de deformación no es un problema trivial, más sin embargo el método más ampliamente aceptado es la afinación del parámetro vía validación cruzada(*cross-validation o CV*). El popular compilador de *data set* en series de tiempo URC *Time Series Archive*, determina para sus *data sets* el valor de *w* al realizar una validación cruzada *leave-one-out CV* con el clasificador 1-NN en los conjuntos de entrenamiento sobre todas las posibles restricciones de ventana, desde 0 hasta el 100 % con incrementos del 1 %, esta técnica elige el tamaño de ventana que maximiza la precisión de entrenamiento, esperando que también se obtenga la mejor precisíon de prueba.

Aún cuando se da nota por parte de los creadores de UCR *Time Series Archive* que esta no podría ser la mejor forma de aprender *w*, es simple, libre de parámetros y además funciona razonablemente bien en la practica, su uso de ha extendido y muchos *papers* utilizan esta aproximación o variantes de ellas.

La técnica que se utlizará para encontrar esta medida se plantea en el *paper*[28], proponiendo una técnica novel que permite el aprendizaje del mínimo valor de *w* para un conjunto de entrenamiento con el que se tiene por objetivo maximizar la calidad en la clasificación en un conjunto de entrenamiento sin etiquetar.

Se plantea en el trabajo que la determinación de un tamaño correcto para *w* produce significativas mejoras en la precisión de clasificación y calidad de agrupamiento, haciendo énfasis en los parámetros que influyen sobre la decisión de esta medida, donde una creencia empírica sin fundamentos como tomar el mayor valor de *w* permitido por los recursos computacionales es la mejor opción, se descarta y comprueba que no es válido para todos los casos, habiendo algunos donde incluso es tan perjudicial escoger un valor pequeño de ventana como uno muy grande, aunque atendiendo a las consecuencias no solo en precisión, sino también en tiempo de cálculo.

Es importante tomar en cuenta que la determinación de un óptimo valor de ventana se encuentra influenciado por el tamaño del conjunto de entrenamiento, la actividad en la que se va a utilizar la medida DTW e incluso la forma de la serie de tiempo. Esto implica que incluso un tamaño de ventana escogido como el maximizador de la precisión para un conjunto de entrenamiento con menos ejemplos, será inválido para el mismo conjunto de entreamiento con una mayor número de elementos. Tampoco es transferible el tamaño de *w* para un proceso de clasificación y otro de agrupamiento.

La técnica se basa en la evaluación de la calidad de clasificación al medir su precisión, y buscando la maximización de la precisión que directamente significa la minimización de la clasificación del porcentaje de error.

Es particularment útil este enfoque pues muestra un aprendizaje robusto para *w* con un conjunto de entrenamiento limitado, utilizando un remuestreo de los datos de entrenamiento. Aún cuando esta técnica no es recomendada en pequeños conjuntos de datos, se subsana este problema remplazando los datos no muestreados con remplazos sintéticos.

Se inicia por como medir la calidad de clasificación. La precisión es una medida de la proporción de resultados verdaderos entre el total de número de casos examinados, multiplicados por 100 para convertirlo en porcentaje. Las métricas utilizadas son: **TP**(*true positive*) y **TN**(true negative), que ambas significan que la etiqueta correcta del patrón corresponde con la etiqueta asignada por el clasificador. Contrario a esto se tiene **FP**(*false positive*) que refiere al número de ejemplos negativos etiquetados como positivos, y **FN**(*false negative*) que refiere al número de ejemplos positivos etiquetados como negativos.

$$Precision = \frac{TP + TN}{TP + TN + FP + FN}$$

Datos sintéticos

La intuición detras de la creación de datos sintéticos. If se remuestrean varios subconjuntos de un conjunto de entrenamiento, se aplica validación cruzada a cada uno y se promedia el resultado de w contra las curvas del porcentaje de error, se espera que este porcentaje imite la curva para el porcentaje de error de prueba y por lo tanto predizca un buen valor para w.

La generación sintética de ejemplares de entrenamientos es plausible, dando la posibilidad de realizar tantas nuevas instancias del conjunto de entrenamiento como se desee, todo con el objetivo de aprender la mejor configuración de w.

Esta producción no es complicada, pues no se requiere la producción sintética de ejemplares que son perfectos en cada aspecto posible, ni que visualmente representen un objeto real para el ojo humano. Es suficiente la creación de objetos con las mismas propiedad con respecto a la mejor configuración para w.

Deformación de ejemplos(*Add warping*)

El añadir una deformación a una serie de tiempo consiste en encojer de forma no lineal una serie de tiempo a una longitud menor al remover de forma aleatoria puntos de datos para después extender linealmente el sub muestreo de serie de tiempo a su tamaño original(Figura 6).

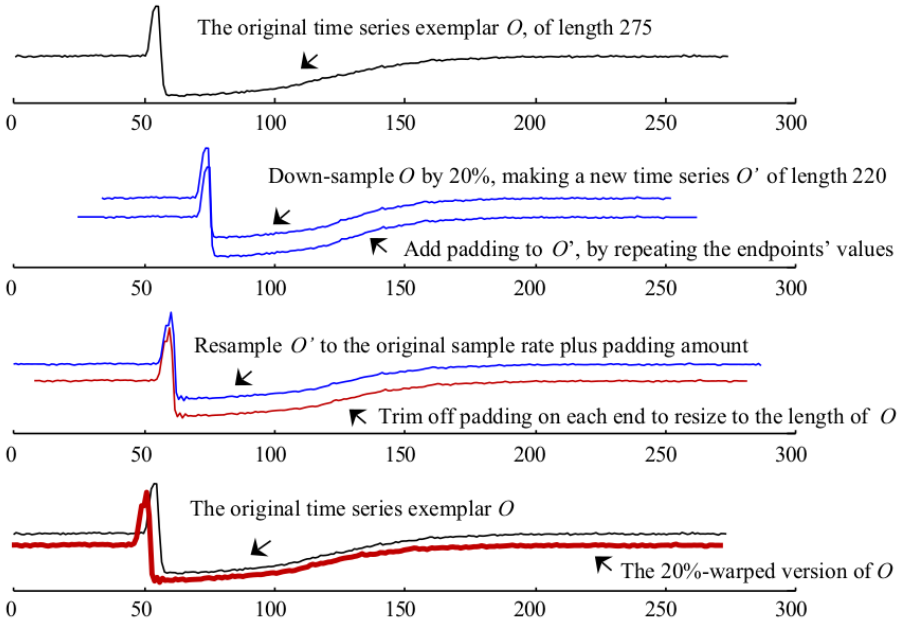

Figura 6: Añadiendo un 20 % de deformación a un ejemplar. Es de notar que en el último panel, la serie de tiempo generada(rojo) is una versión ligeramente deformada de la serie original(negra)
Recuperado de *Adding warping to make new time series*[Figura 24] de Optimizing dynamic time warping's window width for time series data mining applications, 2018

Algoritmo de aprendizaje de *w*

El algoritmo principal se puede resumir en: Realización de N copias del conjunto de entrenamiento original. Para cada copia, remplazar una fracción de los datos con datos generados sintéticamente y realizar una validación cruzada para aprender el procentaje de error contra la curva w. Usando el promedio de todas las curvas N para la predicción de w.

Este algoritmo involucra 3 parámetros:

- **Cantidad de deformación**: A través del estudio empírico de esta función, en el *paper* se describe que la falta de deformación en los elementos provoca que el algoritmo falle. Cuando se añade exactamente la cantidad de deformación necesaria es algoritmo trabaja bien, y en el caso de cuando se añade un exceso de deformación, este sigue funcionado bien, por lo que se decanta en asignar una deformación grande con un 20 %.
- **Relación entre objetos reales y sintéticos**: Igualmente la variación de este parámetro produce diferentes resultados y mas sin embargo en una gran mayoría se produjo un resultado exitoso con un valor de 0.8(mayormente construido por objetos sintéticos).
- ***N*, el número de nuevos conjuntos de entrenamiento**: Este parámetro es sencillo, pues mientras mayor su valor mejor será su resultado, sin embargo en el trabajo se considera un valor de 10.

Finalmente es importante mencionar que esta técnica difiera del típico LOO-CV(Leave-one-out cross validation) para el refinamiento de los parámetros así como lo utiliza el método de UCR *Time Series Archive*, proponiendo un 10-*fold cross validation*, pues provee una buena compensación entre baja varianza LOO y el sesgo a valores grandes de *w* con un 2-*fold* CV.

A pesar de que el uso de 10-*fold* CV pudiera parecer costoso computacionalmente, se pueden acelerar el proceso completo al reducir el número de iteraciones para toda *w*(Incluso 10 iteraciones son suficientes para ofrecer una mejora significativa estadística sobre el método base) o estrechar el rango de valores para *w*.

Modelos con uso de las distancias DTW como características

En el trabajo[14] parten de la declaración del algoritmo DTW como una herramienta excepcionalmente robusta para la medición de distancias entre series de tiempo, y que mediante su combinación con 1-NN se ha conseguido una técnica, que a pesar de ser probablemente el métodos más sencillo es sumamente útil y contundente en su precisión, siendo dificil de superarlo con otras técnicas que atienden la misma tarea.

Con esto en mente, el paper plantea una aproximación distinta donde el algoritmo DTW no es usado únicamente como función de distancia entre series con sus centroides en el modelo de agrupamiento, sino como magnitud capaz de conformar un patrón de nuevas características que posteriormente pueden ser provistos a un modelo estándar de *Machine Learning* mucho más robusto en la tarea de clasificación, tanto para su entrenamiento como más tarde identificación de clase de nuevos patrones de ingreso.

Se asegura que con esta sencilla aproximación se ha logrado mejorar sobre el modelo base DTW 1-NN utilizando ahora un modelo estándar SVM, en 31 de los 47 conjuntos de datos disponibles para *benchmark* de las series de tiempo en UCR. Siendo además importante que el método proveé un mecanismo capaz de combinar métodos basados en distancia DTW con métodos basados en características como SAX, y del que probando su implementación el autor del trabajo prueba una mejora posterior de 37 de los 47 conjuntos de datos de UCR.

Escencialmente el uso de características DTW por si solas no presenta una mejora de desempeño, pero su conjunción con un mejor modelo de clasificación de *Machine Learning* el que mejora su desempeño sobre el método 1-NN.

Vector de características DTW

La idea detras de la conformación de vectores de entrenamiento o clasfificación utilizando la magnitud DTW es en principio sencilla. La representación de una serie de tiempo se va a lograr en términos de las distancias DTW de cada ejemplo de entrenamiento, siendo de esta forma por ejemplo, la primer característica la distancia DTW entre la series de tiempo de entrada con respecto al primer ejemplo de entrenamiento, después la distancia con respecto al segundo ejemplo de entrenamiento resultaría en la segunda característica, etc.

Posterior a la construcción del patrón, se puede utlizar cualquier modelo estándar de *Machine Learning* especializado en la clasificación evitandose así el depender de la etiqueta de clase de la clase más cercana, y permitiendo así al método entrenarse para que aprenda a relacionar la clase de la serie de tiempo a las diferentes distancias DTW de varios ejemplos de entrenamiento.

Tal modelo puede ser por ejemplo, una Maquina de Soporte Vectorial la cual es capaz de manejar correctamente patrones con una gran número de características, además de ser un de los modelos en el *Machine Learning* más robustos y extensamente estudiados.

Combinación de métodos

En adición a la simplicidad de implementación de la aproximación, este método es capaz de extenderse facilmente para ser utilizado en combinación con otro métodos estadísticos o basado en características simbólicas al agregarse estas simplemente como características adicionales en el vector patrón.

Se demuestra en el estudio esta propiedad al combinarse experimentalmente su aproximación con el método *Symbolic Aggregate Approximation*, resultando en un mejor desempeño que cuando ambás técnicas se utilizar de forma independiente e incluso obteníendose resultados que superan al método base en 37 de los 47 conjuntos de datos de UCR cuando se usan en conjunto.

Modelo SAX con comparación de palabras con DTW

En el trabajo[15] se considera la evaluación de 3 métodos útiles para la clasificación de unos patrones motrices derivados de gestos manuales muestreados en sus 3 componentes espaciales del cambio de aceleración de un acelerómetro.

En los 3 métodos se mantiene una clasificación a través de la técnica de agrupamiento no supervisado 1-NN, cambiando escencialmente la función de distancia entre:

- Distancia DTW entre series de tiempo y centroides utilizando la distancia en bloque o L1
- Distancia promedio entre pares de elementos de palabras resultantes de SAX
- Distancia DTW entre palabras resultantes del método SAX para series de tiempo

El primer par de métodos son bastante sencillos, e incluso el primero consiste en el estándar referencial DTW 1-NN. Sin embargo el tercero se muestra como una propuesta interesante que además para la tarea de esta aplicación, resulta obtenerse una mejora en la precisión de los 3 modelos evaluados.

La coexistencia del método SAX y DTW propuesta en el trabajo combina las mejores características de ambos métodos: La eficacia y baja complejidad de SAX, además de la insensibilidad de DTW para las fluctuaciones de velocidad durante la ejecución de los gestos.

En la combinación, se adopta la representación de las series de tiempo mediante una reducción dimensional basada en el alfabeto que proveé el método SAX, sin embargo la inclusión del método DTW se encuentra en la definición de la distancia entre 2 cadenas, implementandose el 'deformamiento' óptimo de la segunda cadena con respecto a la primera, en vez de la comparación por parejas SAX por defecto.

De esta forma la distancia entre las dos cadenas se calcula como la suma de las similitudes de símbolos por pares para todos los pares de símbolos dictados por el resultado de DTW, dividida por la longitud de la cadena(Figura 7).

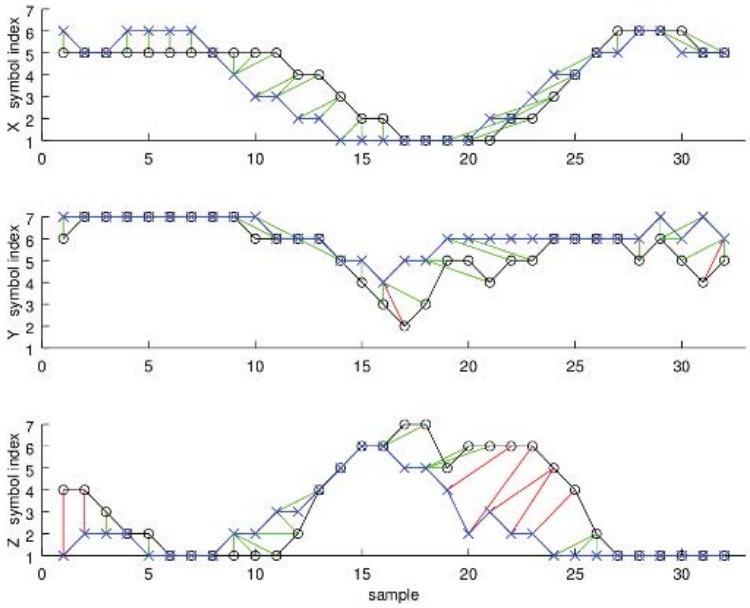

Figura 7: De a cuerdo al método combinado SAX-DTW, únicamente los pares de símbolos los cuales sus índices de símbolo difieren en más de 1 contribuyen en la distancia entre ambas cadenas
Recuperado de *The SAX-DTW method*[Figura 6] de Gesture recognition using Symbolic Aggregate approximation and Dynamic Time Warping on Motion Data.

Técnica propuesta utilizando SAX y DTW

La propuesta de un nuevo modelo útil para la tarea de clasificación de series de tiempo considera las características más importantes del último par de técnicas descritas por los trabajos[14][15]:

- Implementación del algoritmo DTW como magnitud utilizada para la conformación de patrones como característica por su robustez en la determinación de la distancia entre series de tiempo a pesar de la velocidad o desfase entre ambas secuencias.
- Capacidad de tratar exitosamente con series de tiempo ruidosas gracias al modelo SAX
- Eficacia y baja complejidad inherente por el uso del modelo SAX.
- Uso de un modelo estándar clasificador de *Machine Learning* que en conjunto con patrones conformados por características DTW, ofrecen un desempeño superior que el modelo referencial estándar DTW 1-NN.
- Mecanismo capaz de combinar un método basado en distancia como DTW y un método basado en características como SAX.

A prtir de estos el modelo es propuesto como una técnica combinada que utiliza el método SAX para la transformación a un espacio dimensional inferior símbólico, y el algoritmo DTW como función de medida entre la distancia o similitud de 2 series de tiempo para su caracterización.

Partiendo del muestreo del patrón motriz, se obtienen las 3 series de tiempo correspondientes a las mediciones de la fuerza de aceleración sensada en cada uno de los ejes espaciales, conformado de esta forma una triada de series de tiempo. Como acondicionamiento y primer aproximación a la solución, se requerirá de un mecanismo capaz de eliminar tanto al inicio como al final del gesto la acción que permite al prototipo *wearable* muestrear el movimiento.

Con las series de tiempo preprocesadas, se conforma una sola secuencia por facilidad de tratamiento mediante la concatenación del muestreo en el eje x, seguido por la serie de tiempo muestreada en el eje y, y finalmente la secuencia adquirida por la aceleración en el eje z. De esta forma se pasa a conformar 1 sola serie temporal que corresponde a un gesto motriz y no el conjunto de 3 secuencias diferentes en cada eje.

El primer modelo implementado es SAX, con el que se logra transformar la serie de tiempo en una secuencia o cadena símbólica discretizada con la que será más sencillo trabajar, pues además de estandarizar el rango de valores de amplitud que la señal puede tomar, se obliga a que esta nueva cadena tenga la misma longitud de palabras o símbolos, que tienen las demás cadenas utilizadas durante el entrenamiento.

La conformación de un vector como patrón de características DTW, requerirá el cálculo de la similitud de la nueva cadena ingresada con respecto a todas las cadenas representantes de cada clase[27](gestos motrices disponibles para su reconocimiento). De esta manera el patrón resultante será la distancia DTW de la nueva secuencia temporal con respecto a todas las cadenas que representan a cada clase, y con esto finalmente puede alimentarse el modelo clasificador para obtener de regreso la clase a la que corresponde el movimiento del patrón conformado.

El modelo clasificador de *Machine Learning* puede ser cualquier modelo estándar utilizado para este tipo de tareas, SVM multiclase, Softmax, Árbol de decisión, etc. Siendo importante señalar que el entrenamiento del modelo se realiza una vez se hayan calculado las cadenas representantes de cada clase a partir del conjunto de entrenamiento[27], pues precisamente la conformación del patrón requiere su composición mediante el cálculo de la similitud de todos los elementos del conjunto con respecto a los representantes.

El entrenamiento de este modelo se preveé tenga un tiempo de cálculo no insignificante, esto sucede por 3 procesos principalmente:

- Determinación del representante promedio para cada clase de gesto motriz[27]
- Cálculo del tamaño de ventana óptimo w para el algoritmo DTW
- Conformación de c número de matrices de costo acumulado(tantas como clases de gestos existan) por cada elemento del conjunto de entrenamiento

A pesar del tiempo computacional que pueda tomar el entrenamiento del modelo, se hipotetiza un rápido desempeño durante la identificación de cada clase después de la realización de un gesto gracias a la optimización mediante la restricción del tamaño de ventana de la banda Sakoe-Chiba para el algoritmo DTW, siendo el proceso más intensivo computacionalmente. La transformación mediante el modelo SAX y posterior clasificación con cualquier modelo estándar después de haber sido entrenado, no representan una preocupación en el tiempo computacional para su cálculo.

Modelos de *Deep Learning*

El primer y más sencillo modelo de *Deep Learning* es el *Multi Layer Preceptron*(MLP) constituido por una arquitectura completamente conectada de capaz con arreglos de neuronas. Estas conexiones se encuentran modeladas por los pesos de la red neuronal.

Este modelo en específico tiene un impedimento grave con el objetivo de la tarea de clasificación, es ysto es que las MLPs no exiben alguna invariancia espacial, esto se traduce en un propio valor de peso para cada marca de timpo y como resultado la información temporal se pierde, los elementos de la serie de tiempo son tratados independientes unos del otro[16].

Una segund aproximación puede darse con el uso de una arquitectura nombrada *Recurrent Neural Network*(RNN), la cual específicamente utiliza datos secuenciales o series de tiempo, usualmente utilizado para problemas temporales en áreas como traducción de lenguaje, NLP, reconocimiento del habla, etc.

La característica que las distingue y permite naturalmente tratar con secuencias temporales es la 'memoria' con la que cuentan, pues a medida que toman información de entradas anteriores, la entrada y salida actual se encontrará influenciada por esta experiencia previa. Esta arquitectura entonces asume la dependencia de las entradas y salidas, implementando la salida de un estado anterior en la entrada del estado actual como experencia previa.

Y apesar de que este modelo ya es una buena aproximación para tratar con series dependientes del tiempo, tiene un defecto, y es que estas redes no son capaces de recordar dependencias a largo plazo, lo que se traduce en la obtención correcta de resultados cuando se trabaja con secuencias cortas, pero a medida que estas aumentan su longitud los resultados cada vez parecen más aleatorios o sin sentido.

Frente a esto llegamos al primer modelo RNN capaz de manejar problema de la difuminación o desaparición del gradiente, permitiendo entonces la persistencia de la información para secuencias de mayor longitud.

Las redes neuronales recurrentes ***Long Short Term Memory*(LSTMs)**[29], son un primer modelo de *Deep Learning* utilizado popularmente para la clasificación de series de tiempo, y aunque por si sola la arquitectura suele ser suficiente para la predicción precisa de la clase de una serie temporal en aplcaciones no tan complejas, pueden llegar a utilizarse en conjunto con otras arquitecturas como las **Redes Neuronales Convolusionales(CNN)**.

Aunque más comunmente utilizadas en ramas como la Visión por Computadora y Análisis de Imágenes pues permiten la extracción de características en una estructura de matriz generalmente multidimensional, diferentes técnicas se han adaptado para cumplir con la tarea de clasificación de series de tiempo: Una de ellas consiste en aplicar el *kernel* tal como un filtro en una transformación genérica no lineal sobre la serie de tiempo, y repitiendo este proceso de aplicación de varios filtros, resulta en una serie de tiempo donde las dimensiones son iguales al número de filtros usados y que la intuición detrás es aprender multiples características discriminantes útiles para la tarea de clasificación[16]. Otro enfoque sería la transformación de la secuencia temporal a una imagen, utilizando la conversión a un espectograma, o el uso de grafos recurrentes como visualización de la estructura recurrente de una serie de tiempo[30].

Una alternativa de modelo especialmente evaluado prácticamente con archivo UCR/UEA para la clasificación de secuencias temporales, es la arquitectura ***Fully Convolutional Neural Network*(FCNs)**. La característica principal de esta arquitectura de red convolusional es la inexistencia de capas locales de *pooling*, permitiendo la permanezca inalterada la longitud de la serie de tiempo a través de las convoluciones[16]. Así también en adición, la arquitectura remplaza la tradicional capa final FC con una capa *Global Average pooling*(GAP) que reduce drásticamente el número de parámetros en una red mientras habilita el uso de CAM que resalta que partes de la entrante serie de tiempo contribuye más a cierta clasificación.

Una cuarta arquitectura propuesta es la **Red Residual(ResNet)**, siendo una arquitectura profunda de 11 capas, de las cuales 9 son convolucionales seguidas por una capa GAP que promedia las series de tiempo a través de la dimensión temporal. La principal característica de esta red es la conexión residual atajo entre capas convolusionales consecutivas, habilitando el flujo del gradiente directamente a través de estas conexiones, lo que permite el entrenamiento de la DNN mucho más sencillo por la reducción del efecto de desaparición del gradiente.

ROCKET[31][32]

Se revisa como último modelo una arquitectura que tiene como objetivo ser rápida y precisa en la tarea de la clasificación de secuencias temporales, así mismo de todas las técnicas estudiadas hasta el momento se trata de la más actual. **ROCKET(*RandOM Convolutional KErnel Transform*)** es un método que se jacta de conseguir el mismo nivel de precisión en la tarea de clasificación, pero en tan solo una fracción del tiempo en competencia con algoritmos estados del arte, incluyendo CNNs.

ROCKET en primer instancia transforma la serie de tiempo utilizando *Kernel* de convolución aleatorios, como los usados en CNNs, y entonces entrena un clasificador lineal con estas características. Este método es capaz de extraer con un único mecanismo un gran número de mismas características que aquellas técnicas en las que se depende de una representación específica que transforma la serie de tiempo en otro dominio espacial, tales como forma, frecuencia o varianza.

Su característica probablemente más atractiva es el tiempo que requiere para su entrenamiento, de tan solo 1 hora y 40 minutos para la misma tarea que el siguiente mejor tiempo con el modelo cBOSS le tomó 19 horas y 33 minutos[].

Kernels Convolusionales

Los *Kernels* igual a los encontrados en CNN son inicializados con valores aleatorios para longitud, pesos, **bias**, *dilation* y *padding*. ROCKET utiliza un número muy grande de *kernels*, por defecto 10,000, siendo posible esta cantidad por el bajo costo computacional y esto se debe a el hecho de que los pesos del *kernel* no son 'aprendidos' y existen tan solo 1 capa de convoluciones.

La asignación aleatoria de los parámetros del *kernel* permite al método capturar un amplio rango de información, en particular la variedad en el valor de *dialation* en el *kernel* le permite a ROCKET caputar patrones en diferentes frecuencias y escalas.

La transformación del *kernel* Convolucional

Cada *kernel* convoluciona con cada serie de tiempo para producir un mapa de características. Este mapa después es 'agregado(*aggregated*)' para producir 2 características por kernel:

- El **valor máximo**: Similar a *global max pooling*
- La **razón de valores positivos**: Indica como pesar la prevalencia de un patrón capturado por el *kernel*. Este valor es el elemento más crítico de ROCKET que contribuye a su alta precisión(Figura 8)

$$ppv = \frac{1}{n}\sum_{i=0}^{n-1}[z_1 > 0]$$

Figura 8: Donde z_i es la salida de la operación convolucional.

Clasificación Lineal

Para conjuntos de datos pequeños, se recomienda un Clasficador de Regresión de Cresta debido a su rápida validación cruzada de la regularización paramétrica y no otro hiperparámetro.

La regularización es crítica cuando el número de características excede el número de ejemplos de entrenamiento, como suele ser el caso para conjuntos de datos pequeños.

En el caso de conjuntos de datos grandes se recomienda Regresión Logística con gradiente estocástico descendente debido a la escalabilidad. En estos conjuntos de datos grandes el número de ejemplos de entranemiento es mucho más grande que el número de características extraidas.

7. Discusión y conclusiones

La tarea de clasificación de series de tiempo es una antigua y de gran influencia en muchas de las ramas humanas donde naturalmente se realiza una descripción del muestreo de los datos en una estructura secuencial dependiente del tiempo, y precisamente resultado de este interesante problema, durante décadas se han realizado trabajos de investigación y desarrollo enfocadas en proponer nuevas alternativas de modelos que sean capaces de alguna u otra forma, superar alguno de los aspectos mejorables en antiguas técnicas(en precisión, simplicidad, transformación del espacio de características, desempeño, tiempo de cálculo, etc).

Desde un principio la dominancia del algoritmo DTW para la similitud entre series de tiempo y el sencillo modelo no supervisado 1-NN fue clara, habiendo poco espacio para nuevos modelos capaces de superar la precisión de este estándar, teniendo poco a poco avances mediante la optimización de las restricciones en el algoritmo DTW, transformación del espacio de características en otro distino al temporal, uso de diferentes funciones de medidas a la Euclideana y DTW, conjunción de métodos basados en distancias y métodos basados en características, más tarde la conjunción de ensamblados con diferentes tipos de clasificadores en diferentes configuraciones de transformación, y finalmente con modelos de *Deep Learning* cada vez más complejos que inician a tomar mas relevancia con el paso del tiempo por sus características y posibilidades frente a los antiguos métodos de clasificación.

Del descubrimiento de estos tantos métodos, el enfoque de interés principal se mantuvo en métodos ciertamente más recientes al base DTW 1-NN, pero no contemporáneos como las aproximaciones en *Deep Learning*. La razón de esto se encuentra principalmente en la arquitectura física del sistema embebido en el que se planea implementar en un futuro durante su desarrollo el modelo clasificador de los gestos motrices.

La estructura embebida de la solución propuesta se traduce en mayor portabilidad con un detrimento en el poder computacional y capacidades hardware. Esto tienen una implicación directa en el tipo de modelo a ser utilizado durante la etapa de clasificación, pues si bien podría ser para el caso de la técnica propuesta una etapa de entrenamiento pesada, este proceso del modelo puede realizarse en un equipo con mayores prestaciones computacionales.

Esta propuesta de modelo realizada que combina el modelo SAX para la transformación de dimensionalidad a cadenas de palabras, con el algoritmo de similitud entre secuencias temporales DTW, es planteado de manera teórica en este documento de investigación, por lo que ofrece la oportunidad futura del desarrollo de un documento enfoca en mostrar los resultados empíricos de la implementación experimental del modelo con respecto a las demás técnicas revisadas.

Así mismo como trabajo futuro en una alternativa interesante y posiblemente adaptable a la arquitectura del sistema embebido de la propuesta, se encuentra el modelo de *Deep Learning* ROCKET en su versión simplificada y aún más veloz nombrada **MiniRocket**[33], la cual asegura mantener un margen aceptable de precisión frente a otros modelos del estado del arte y el propio ROCKET, con una actualizada y optimizada complejidad temporal computacional para el entrenamiento del modelo de clasificación.

La posible o no adptación de este nuevo modelo en la solución tan solo podría definirse mediante una implementación experimental de la problemática con la técnica en la arquitectura y componentes disponibles del sistema embebido final propuesto para el Trabajo Terminal.

8. Referencias

[1] N. Marques, "¿Qué es el lenguaje?,"*Babbel*, 02, 2018 [En línea]. Disponible en https://es.babbel.com/es/magazine/que-es-lenguaje

[2] O. Castillero Mimenza, "Los 8 tipos de trastornos del habla,"*Psicología y Mente*. [En línea]. Disponible en https://psicologiaymente.com/clinica/tipos-trastornos-habla

[3] National Institute on Deafness and Other Communication Disorders,(2017, 03. 06). "La afasia".[En línea]. Disponible en https://www.nidcd.nih.gov/es/espanol/afasia

[4] A. Triglia, "Afasias:los principales trastornos del lenguaje,"*Psicología y Mente*. [En línea]. Disponible en https://psicologiaymente.com/clinica/afasias-trastornos-lenguaje

[5] "Apraxia", *Instituto Nacional de Trastornos Neurológicos y Accidentes Cerebrovasculares*. 03, 2022.[En línea]. Disponible en https://espanol.ninds.nih.gov/es/trastornos/apraxia

[6] J. Huang, "Apraxia,"*MSD*. 10, 2021.[En línea]. Disponible en https://www.msdmanuals.com/es-mx/hogar/enfermedades-cerebrales,-medulares-y-nerviosas/disfunci %C3 %B3n-cerebral/apraxia

[7] "La Disartria,"*American Speech Language Hearing Association*.[En línea]. Disponible en https://www.asha.org/public/speech/Spanish/La-Disartria/

[8] J. Huang, "Disartria,"*MSD*. 10, 2021.[En línea]. Disponible en https://www.msdmanuals.com/es-mx/hogar/enfermedades-cerebrales,-medulares-y-nerviosas/disfunci %C3 %B3n-cerebral/disartria

[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] Amidon, A., 2022. How to Apply K-means Clustering to Time Series Data. [online] Medium. Available at: ¡https://towardsdatascience.com/how-to-apply-k-means-clustering-to-time-series-data-28d04a8f7da3¿[Accessed 6 May 2022].

[14] 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

[15] Mezari, Antigoni & Maglogiannis, Ilias. (2017). Gesture recognition using symbolic aggregate approximation and dynamic time warping on motion data. 342-347. 10.1145/3154862.3154927

[16] 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

[17] C. Chatfield, "Introduction," *Analysis of time series an Introduction*, 5. Reino Unido: Chapman & Hall, 1996, pp. 11-15.

[18] D. Peña, G. C. Tiao, R. S. Tsay, "Introduction," *A Course in Time Serie Analysis*. New York: J. Wiley, 2001, pp. 1.

[19] M Müller. "Dynamic Time Warping," *Information Retrieval for Music and Motion*, 4. Alemania: Springer, 2007, pp. 69-73.

[20] 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

[21] Tan, Chang Wei & Herrmann, Matthieu & Forestier, Germain & Webb, Geoffrey & Petitjean, François. (2018). Efficient search of the best warping window for Dynamic Time Warping.

[22] Brownlee, J., 2022. How to Normalize and Standardize Time Series Data in Python. [online] Machine Learning Mastery. Available at: ¡https://machinelearningmastery.com/normalize-standardize-time-series-data-python/¿[Accessed 9 May 2022].

[23] Krishnamoorthy, V., 2022. Piecewise Aggregate Approximation. [online] Vigne.sh. Available at: ¡https://vigne.sh/posts/piecewise-aggregate-approx/¿[Accessed 10 May 2022].

[24] Krishnamoorthy, V., 2022. Symbolic Aggregate Approximation. [online] Vigne.sh. Available at: ¡https://vigne.sh/posts/symbolic-aggregate-approximation/¿[Accessed 10 May 2022].

[25] 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].

[26] 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

[27] S. Datta, C. K. Karmakar and M. Palaniswami, .Averaging Methods using Dynamic Time Warping for Time Series Classification,"2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, pp. 2794-2798, doi: 10.1109/SSCI47803.2020.9308409.

[28] 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

[29] Education, I., 2022. What are Recurrent Neural Networks?. [online] Ibm.com. Available at: ¡https://www.ibm.com/cloud/learn/recurrent-neural-networks¿[Accessed 11 May 2022].

[30] Kaggle.com. 2022. RecuPlots and CNNs for time-series classification. [online] Available at: ¡https://www.kaggle.com/code/tigurius/recuplots-and-cnns-for-time-series-classification/notebook¿[Accessed 11 May 2022].

[31] Amidon, A., 2022. ROCKET: Fast and Accurate Time Series Classification. [online] Medium. Available at: ¡https://pub.towardsai.net/rocket-fast-and-accurate-time-series-classification-f54923ad0ac9¿[Accessed 11 May 2022].

[32] Dempster, A., Petitjean, F. & Webb, G.I. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min Knowl Disc 34, 1454–1495 (2020). https://doi.org/10.1007/s10618-020-00701-z

[33] Amidon, A., 2022. MiniRocket: Fast(er) and Accurate Time Series Classification. [online] Medium. Available at: ¡https://medium.com/towards-data-science/minirocket-fast-er-and-accurate-time-series-classification-cdacca2dcbfa¿[Accessed 11 May 2022].