

Relazione per la prova finale
«Implementazione e Ottimizzazione dell'Algoritmo
Dynamic Time Warping»

Relatore: Prof. Gian Antonio Susto

Laureando: Matteo Piran
Matricola n.: 2103432

Padova, 22/09/2026

Machine Learning

Branca dell'Intelligenza Artificiale che si occupa dello sviluppo di algoritmi e tecniche finalizzate all'apprendimento automatico mediante la statistica computazionale e l'ottimizzazione matematica.

I casi d'uso comuni di machine learning nelle aziende includono l'identificazione e la classificazione degli oggetti, il rilevamento delle anomalie, l'elaborazione dei documenti e l'analisi predittiva¹.

Gli algoritmi per l'identificazione e la classificazione analizzano dati di input per riconoscere oggetti, eventi o pattern e assegnarli a categorie predefinite.

K-Nearest Neighbors (k-NN)

È un semplice approccio ai problemi di classificazione dove un nuovo elemento non ancora classificato viene inserito nella categoria che risulta più frequente tra i suoi k elementi più vicini.

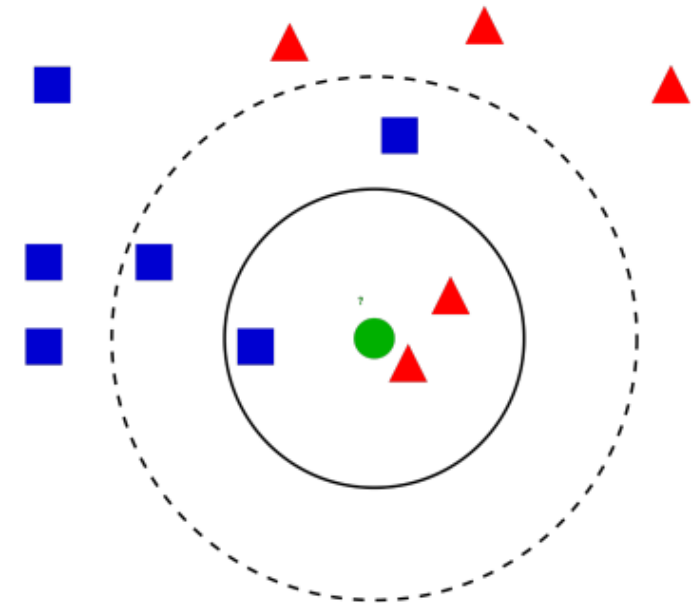
Green: new data point

Blue and **Red:** historical data of 2 different classes

$k = 1$, classified as **Red**

$k = 3$, classified as **Red**

$k = 5$, classified as **Blue**



Analisi di Serie Temporal

L'Analisi di Serie Temporal è una branca del ML che si interessa di analizzare segnali sul tempo con l'obiettivo di trovare trend, riconoscere pattern, o in genere di cercare di arrivare a fare delle predizioni in base ai dati ricevuti.

Un'idea semplice per eseguire un'analisi tra due segnali sul tempo valutandone in qualche modo un "coefficiente di somiglianza" potrebbe essere quella di valutare la distanza tra i due campioni allo stesso istante di campionamento (s_1-t_1) e di sommarlo alla distanza di ciascun altra coppia di valori, questo valore che andiamo così a trovare è detto Distanza Euclidea:

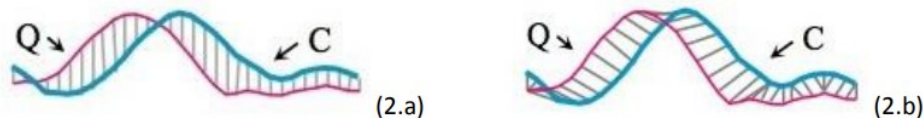
$$DE = \sum_{i=0}^n |s_i - t_i|$$

Analisi di Serie Temporali

Un problema evidente della Distanza Euclidea è il fatto che non è in grado di riconoscere ritardi o accelerazioni o piccole variazioni locali tra due segnali sul tempo altresì molto simili.

Risulta dunque chiaro che in moltissimi casi dove abbiamo degli sfalsamenti lungo l'asse del tempo o comunque due segnali non abbastanza "puliti" e ben allineati l'approccio risolutivo della Distanza Euclidea non sarà sufficiente a farci raggiungere dei risultati soddisfacenti.

È per questo che opteremo per un approccio basato su Programmazione Dinamica.



Valutando i due grafici con la Distanza Euclidea risultano molto diversi; con la programmazione dinamica vengono associate le similitudini anche se sfalsate nel tempo

Fig.2 «Data Mining per Serie Storiche», Gennaro, 2013

OBIETTIVO DEL LAVORO

Implementazione Dynamic Time Warping

L'obiettivo di questo lavoro è l'implementazione da zero dell'algoritmo Dynamic Time Warping (DTW), basandoci sulla formalizzazione originale introdotta da Sakoe e Chiba² e testare la sua performance in problemi di classificazione e osservare su cosa possiamo agire per ottimizzarne il funzionamento.

Dopo una prima implementazione basica e funzionante osservando degli evidenti limiti alla velocità di calcolo dell'algoritmo abbiamo aggiunto la Sakoe-Chiba Band, strumento noto in letteratura² per diminuire sensibilmente la quantità di calcoli svolti dall'algoritmo e, inoltre, in grado di ridurre i casi di Overfitting dovuti ad un'eccessiva libertà nel confronto dinamico tra due segnali.

OBIETTIVO DEL LAVORO

Ottimizzazione Dynamic Time Warping

Sebbene l'algoritmo DTW sia stato originariamente formulato per il riconoscimento di parole parlate, la sua applicazione è stata estesa con successo all'analisi, al data mining e alla ricerca di pattern in serie temporali generiche³, come abbiamo fatto anche nel nostro caso.

La restante parte del lavoro si è quindi concentrata sull'osservare come la variazione dell'ampiezza della banda di Sakoe-Chiba non solo comportasse un'enorme variazione nel tempo di esecuzione ma anche nelle performance dell'algoritmo su diversi dataset.

Abbiamo dunque cercato di individuare quali fossero i valori per ottimizzare DTW su tre diversi dataset dall'archivio UCR/UEA Time Series Classification Archive⁴, ovvero GunPoint, ECG200 e Trace.

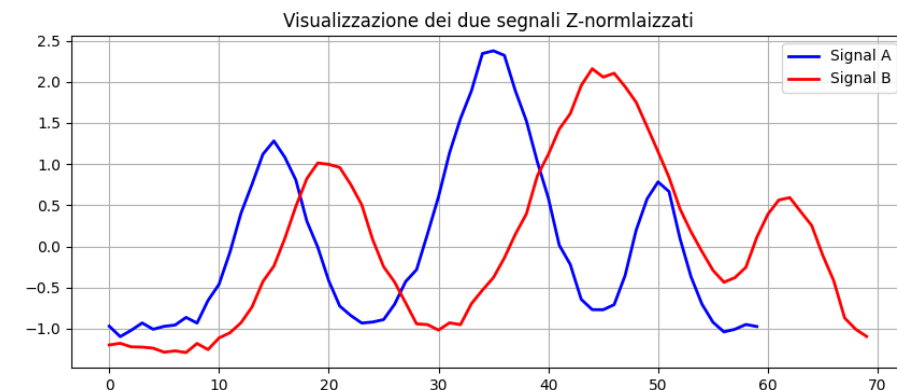
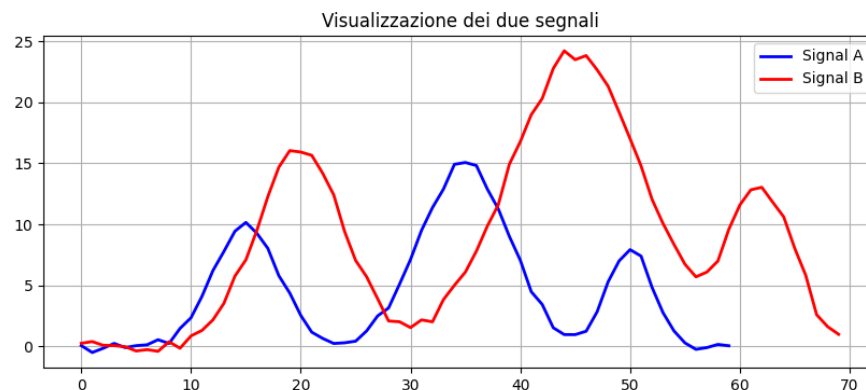
DTW Normalizzazione

Analizziamo come funziona DTW e come arriva a restituire il "coefficiente di somiglianza" tra due segnali A e B.

Per cominciare, l'algoritmo riceve due segnali in input e procede a **Z-normalizzarli** in modo che la procedura risulti efficace anche su segnali scalati in modo diverso.

$$Z = \frac{X - \mu}{\sigma}$$

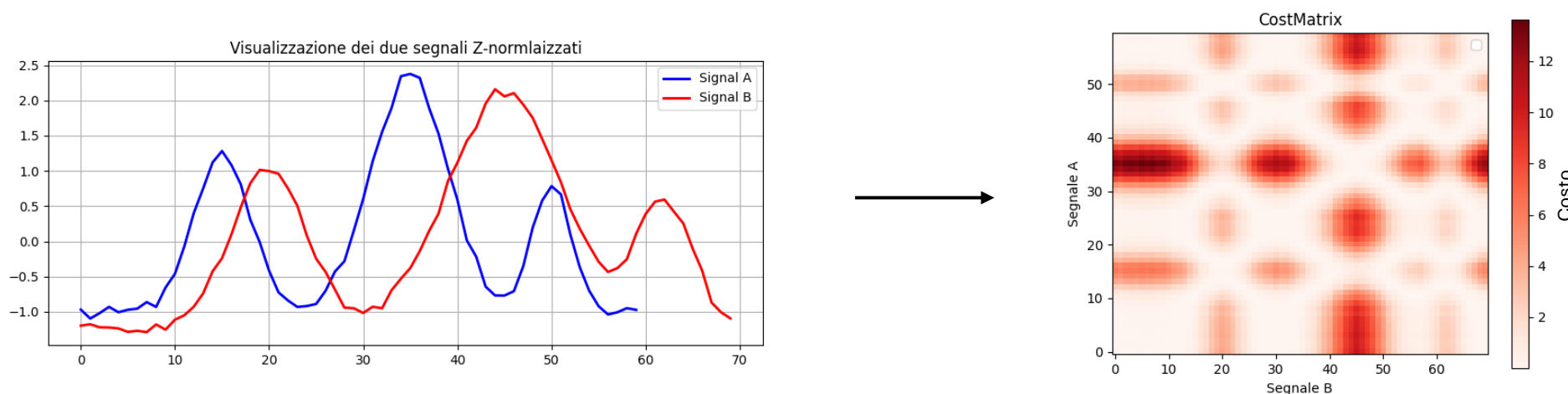
Il secondo grafico rappresenta i due segnali Z-normalizzati, infatti risultano entrambi con media e varianza =1; la normalizzazione modifica i valori solo lungo l'asse delle y.



DTW Cost Matrix

A differenza della Distanza Euclidea, che impone un confronto rigido "uno a uno" nello stesso istante di tempo, il DTW deforma l'asse temporale in modo elastico per consentire associazioni "molti a uno".

Per raggiungere questo obiettivo, ignoriamo il vincolo del tempo e calcoliamo la distanza tra ogni singolo campione del primo segnale e tutti i campioni del secondo. Questi valori vengono rappresentati nella **Cost Matrix**: una mappa che ci mostra visivamente e numericamente quali coppie di punti hanno la distanza, e quindi il costo, minore.

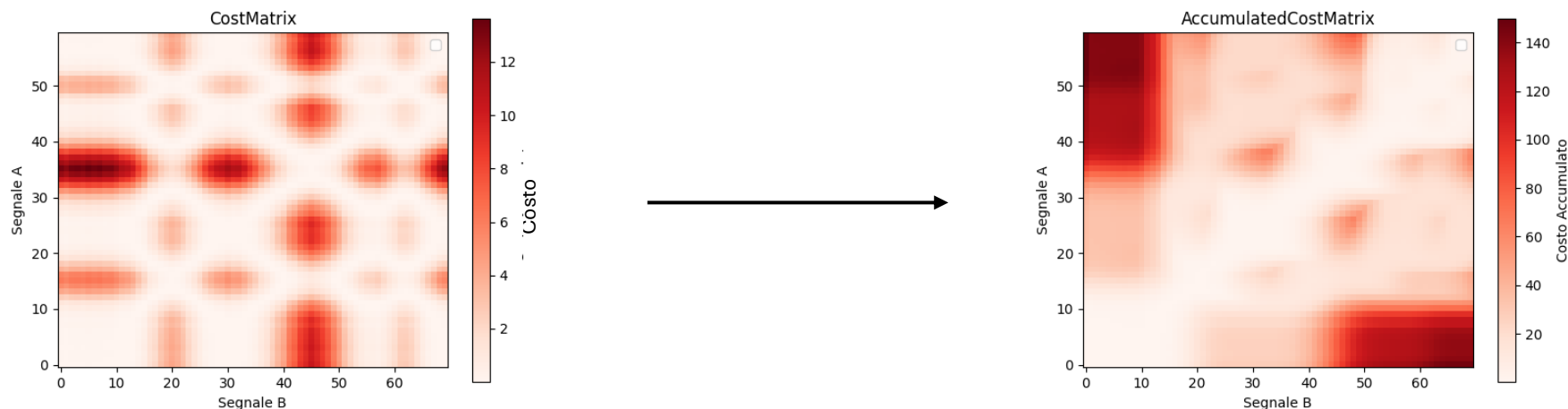


DTW Accumulated Cost Matrix

Per risolvere i problemi di allineamento ci interessiamo del Costo Cumulativo più che del Costo Locale: creiamo quindi l' **Accumulated Cost Matrix (ACM)** che non valuta più la singola distanza isolata, ma il costo minimo totale necessario per raggiungere una specifica posizione.

Sfruttiamo la programmazione dinamica per scomporre la ricerca del percorso globale in sotto-problemi sequenziali dove troviamo il valore di ogni coordinata (i, j) come la somma della distanza locale più il minimo dei costi cumulativi già calcolati nelle celle precedenti adiacenti.

$$ACM(i, j) = \min(ACM(i - 1, j); (i, j - 1); (i - 1, j - 1)) + CM(i, j)$$



DTW Warping Path

Una volta creata la ACM osserviamo il valore nell'angolo in alto a destra della matrice alla coordinata (n, m) quello sarà il valore del "coefficiente di somiglianza" tra i due segnali.

Ricostruendo attraverso l'ACM il percorso che porta da $(0, 0)$ a (n, m) passando per i costi minimi troviamo il **Warping Path** che indica l'allineamento ideale identificato dall'algoritmo per massimizzare la somiglianza dei due segnali, che graficamente ci permette di osservare le coppie di segnali nell'allineamento trovato da DTW.

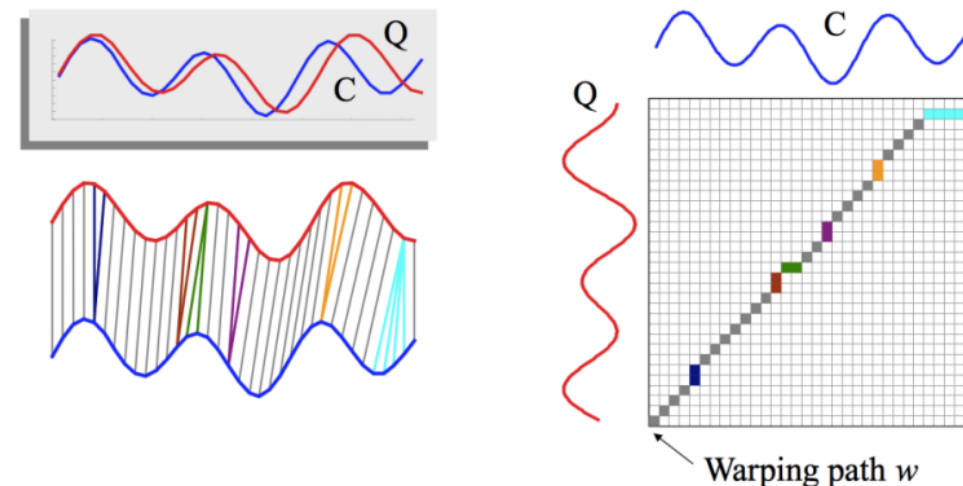
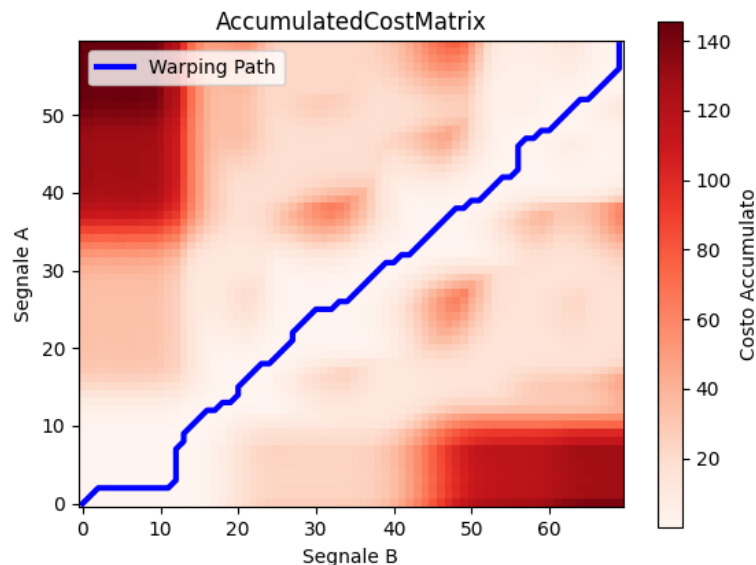
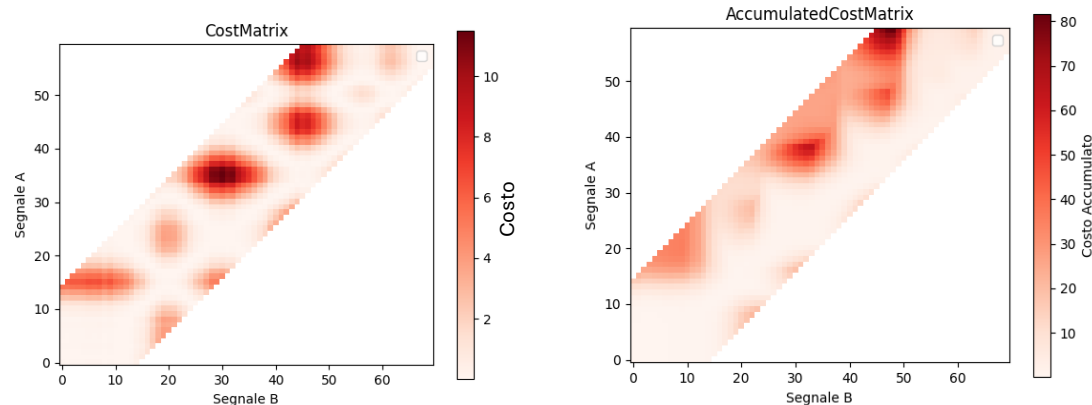


Fig. 3 «Data Mining per Serie Storiche», Gennaro, 2013

DTW Sakoe-Chiba Band

Il calcolo dell'intera matrice dei costi cumulativi (ACM) comporta un costo computazionale notevole ($\Theta(n^2)$). Inoltre, esplorare soluzioni di allineamento troppo lontane dalla diagonale principale risulterebbe controproducente: si otterrebbero associazioni tra campioni temporalmente troppo distanti, perdendo il significato fisico dell'allineamento.

Per questi motivi, si sceglie di implementare la **Sakoe-Chiba Band**², un vincolo globale che limita i calcoli dell'algoritmo entro una specifica finestra temporale.



Sia la CM che la ACM vengono limitate dalla banda, l'algoritmo è costretto a creare il Warping Path all'interno di questo limite

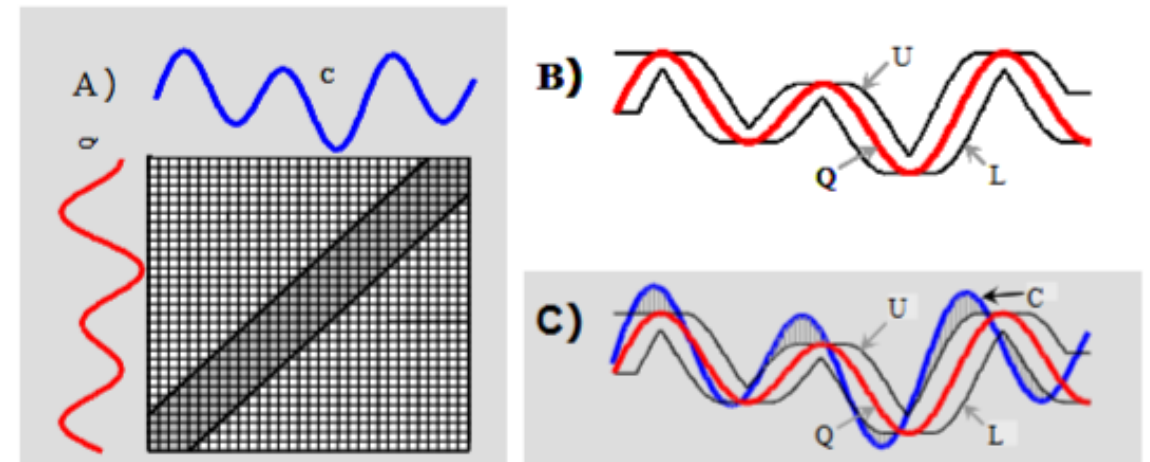


Fig. 3 "Everything you know about Dynamic Time Warping is wrong" Ratanamahatana, Keogh, 2004

Testing: i dataset

A questo punto, disponendo di una versione funzionante di DTW, ho cercato di sperimentarne il funzionamento utilizzando tre dataset appartenenti al database UCR/UEA Time Series Classification Archive⁴:

- **GunPoint**: basato sul movimento della mano destra di due attori. La classificazione chiede di distinguere il gesto di estrarre una finta arma dalla fondina e puntarla rispetto al semplice gesto di puntare il dito indice verso un bersaglio.
- **ECG200**: raccoglie tracce elettrocardiografiche in cui ogni serie temporale rappresenta l'attività elettrica del cuore durante un singolo battito. La classificazione mira a distinguere i battiti cardiaci normali da quelli anomali.
- **Trace**: progettato per simulare il monitoraggio di segnali strumentali all'interno di una centrale nucleare. L'obiettivo è classificare diverse tipologie di anomalie transitorie (eventi o guasti) che possono verificarsi in istanti temporali differenti e non prevedibili.

Testing: l'approccio

La valutazione dell'algoritmo sui vari dataset è stata effettuata utilizzando un classificatore 1-NN.

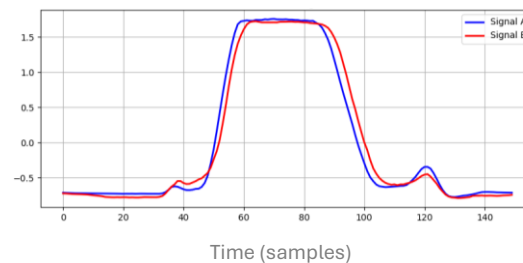
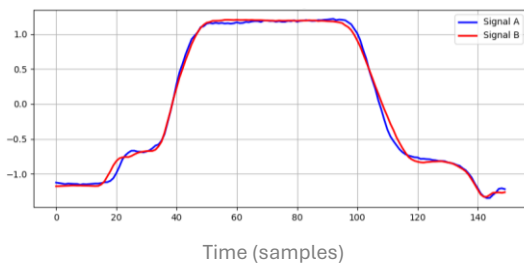
Per ogni campione del test set è stato calcolato il costo del Warping Path rispetto a tutti gli elementi del training set, per poi assegnargli la classe dell'elemento di addestramento associato al costo minimo (massima somiglianza).

Testing: GunPoint

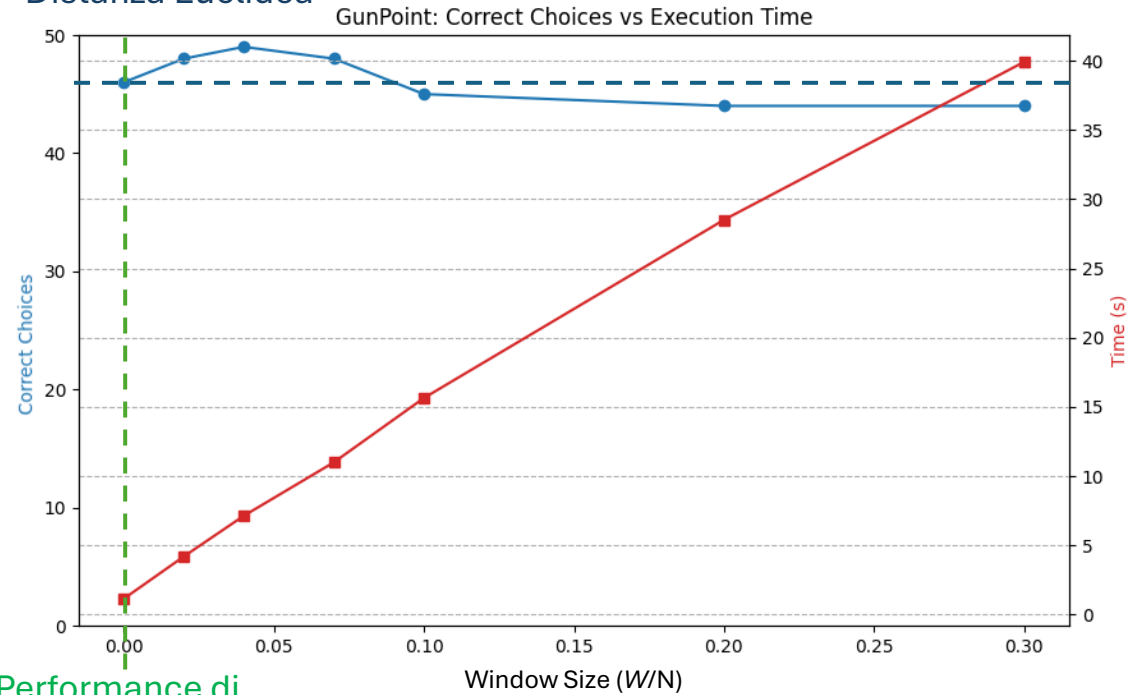
I risultati sul dataset GunPoint sono stati molto utili: DTW performa meglio della distanza euclidea (49/50 contro 46/50) con i valori di W compresi tra 2% e 7% con un massimo al 4%.

Tuttavia le performance scendono al di sotto della distanza euclidea dopo che la finestra supera il 7% a causa dell'Over-Warping (osserveremo che anche il dataset ECG200 risente di questo fenomeno).

Due segnali di GunPoint correttamente associati da DTW



Correct Choices di
Distanza Euclidea



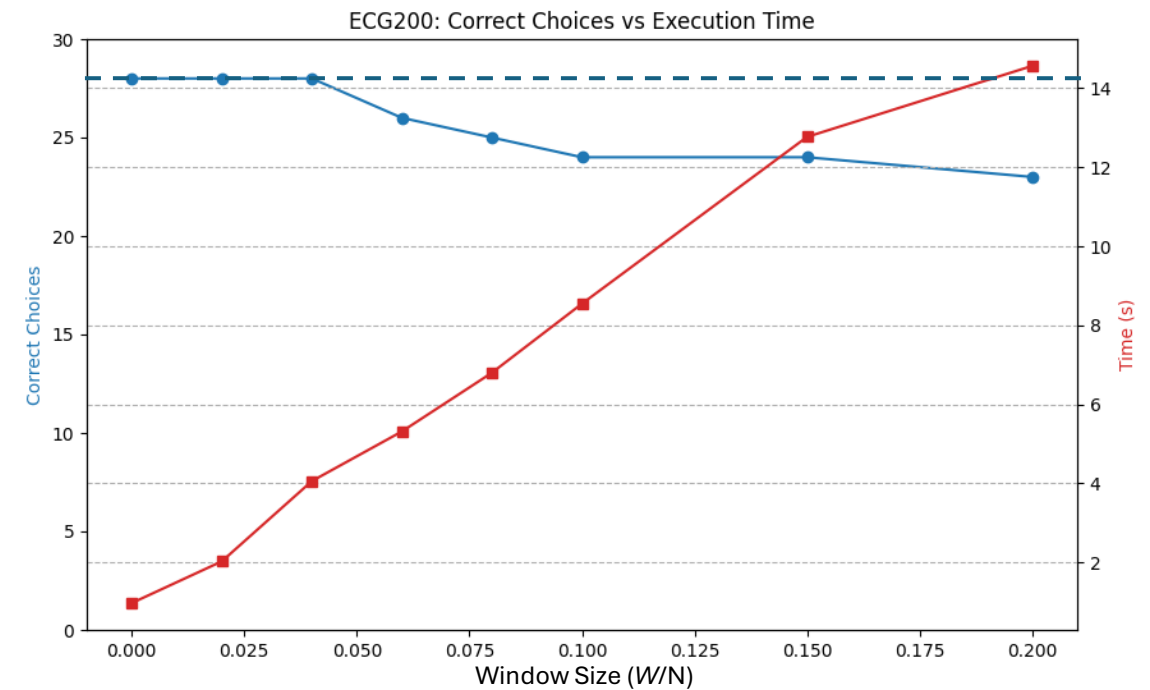
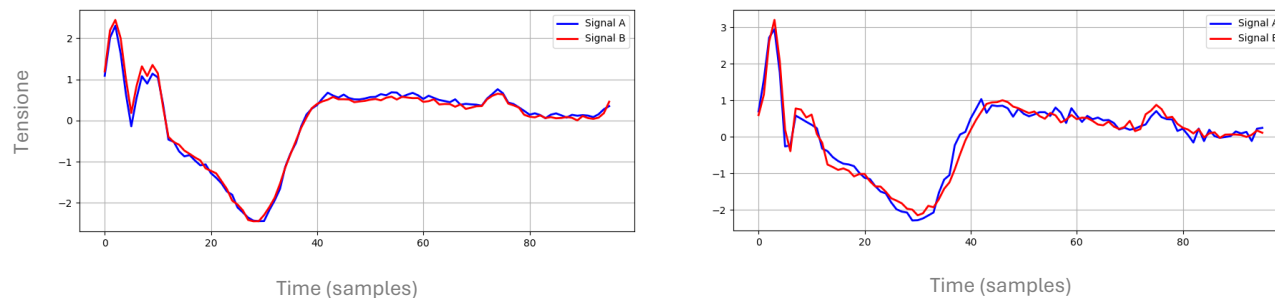
Performance di
Distanza Euclidea

50 test con W 0%-30%

Testing: ECG200

I risultati sul dataset ECG200 sembrano molto simili a quelli di GunPoint senza però avere nessun vantaggio ad usare DTW poiché con nessun valore di W viene superato il punteggio della distanza euclidea (27/30).
Nuovamente osserviamo che dopo che la finestra supera il 5% abbiamo un significativo calo nelle performance di DTW a causa dell'Over-Warping.

Due segnali di ECG200 correttamente associati da DTW

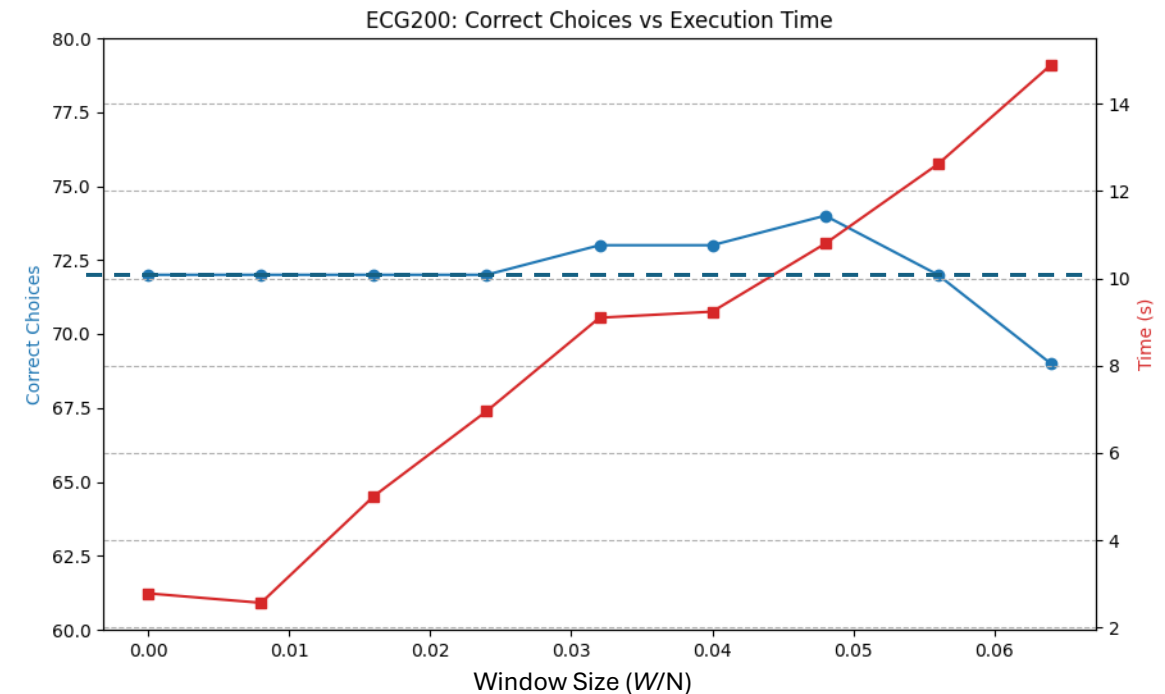


30 test con W 0%-20%

Testing: ECG200

Svolgendo però più test e riducendo a 0.8% gli incrementi delle W sperimentate osserviamo che la gaussiana che avevamo trovato nel dataset GunPoint ricompare, mostrandoci un valore di picco a $W=4.8\%$, con il quale DTW arriva a performare meglio della distanza euclidea (74/80 vs 72/80) impiegando però più di 10 volte più tempo durante l'esecuzione dei test.

Ciò dimostra che anche su questo dataset DTW è in grado di performare meglio della distanza euclidea, con la necessità di una scelta ben studiata del parametro W .



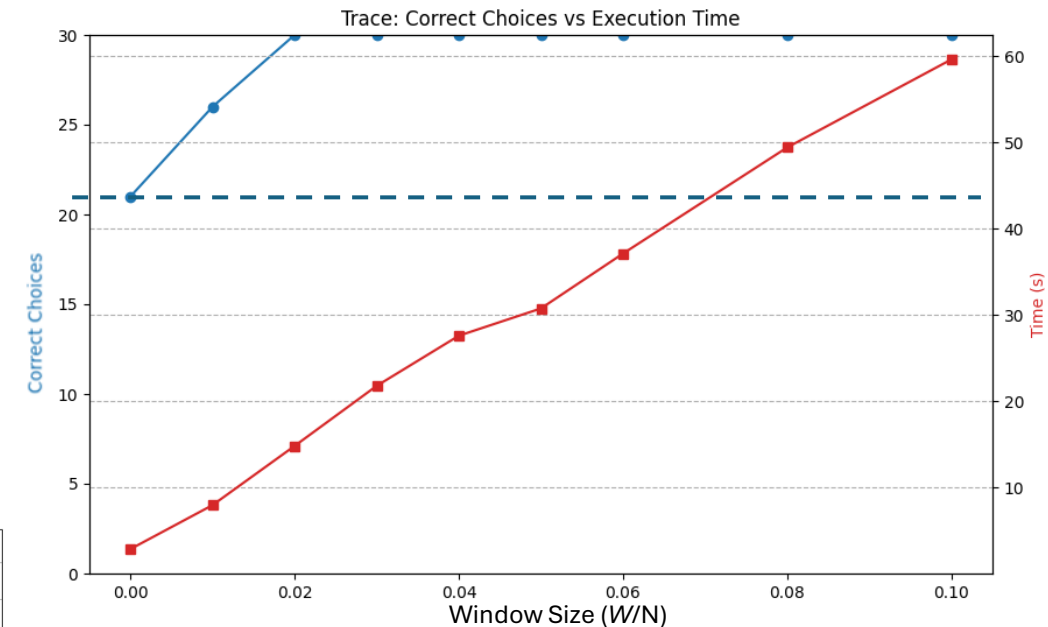
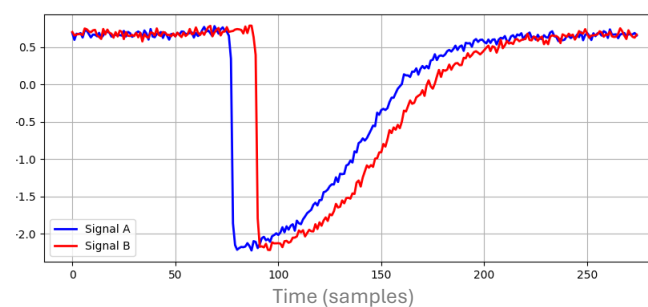
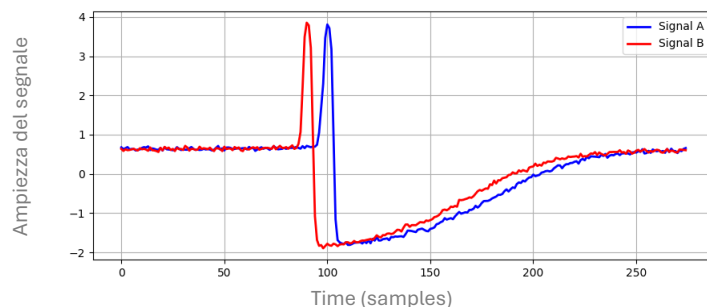
80 test con W 0%-6,4%

Testing: Trace

I risultati sul dataset Trace si presentano con un grafico molto diverso rispetto ai precedenti: osserviamo che DTW performa al 100% di precisione con W superiore al 2%.

Nel test con la Distanza Euclidea i risultati sono notevolmente più scarsi (21/30) e sembra che al crescere della finestra non appaia Over-Warping.

Due segnali di Trace correttamente associati da DTW

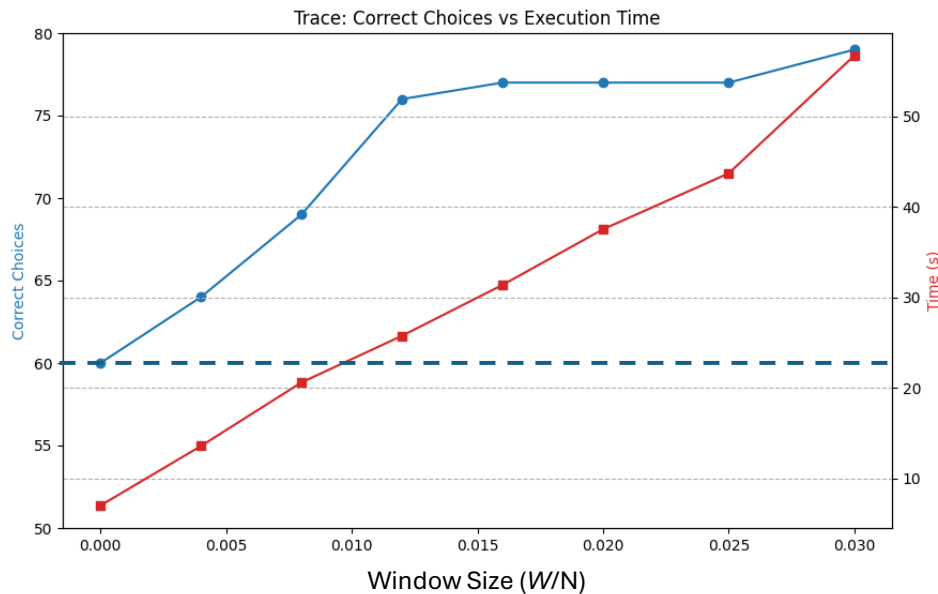


30 test con W 0%-10%

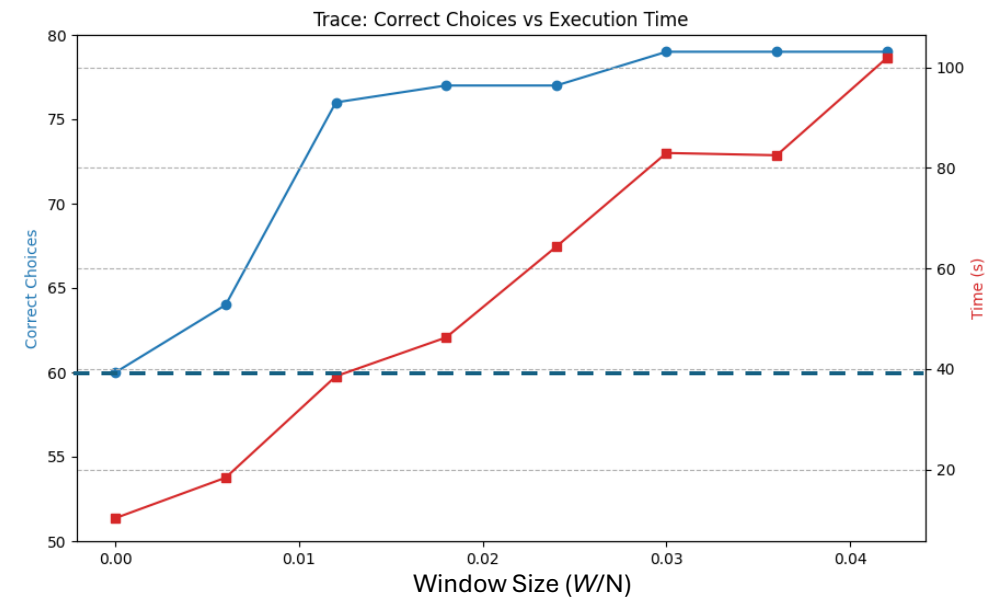
Testing: Trace

L'obiettivo è stato identificare il valore minimo di W in grado di massimizzare l'accuratezza riducendo al minimo il costo computazionale.

L'analisi empirica sul dataset ha individuato questo *trade-off* ottimale al valore $W= 3\%$.



80 test con W 0%-3%



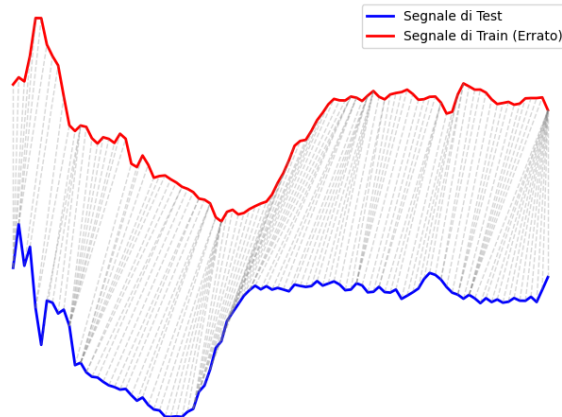
80 test con W 0%-4,2%

Over-Warping

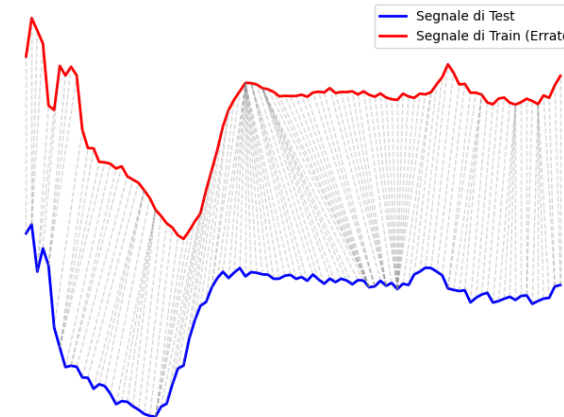
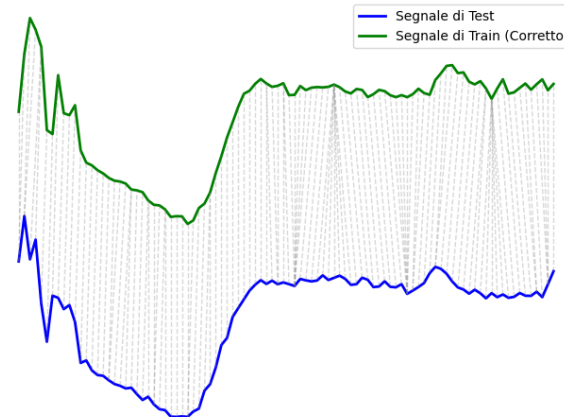
Con Over-Warping in letteratura⁵ si descrive il fenomeno di Overfitting dovuto ad un'ampiezza della finestra di Sakoe-Chiba troppo grande.

In molti casi è importante, per il funzionamento migliore della classificazione, che DTW non sia troppo concessivo nel caso di coppie di funzioni molto sfalsate sull'asse temporale.

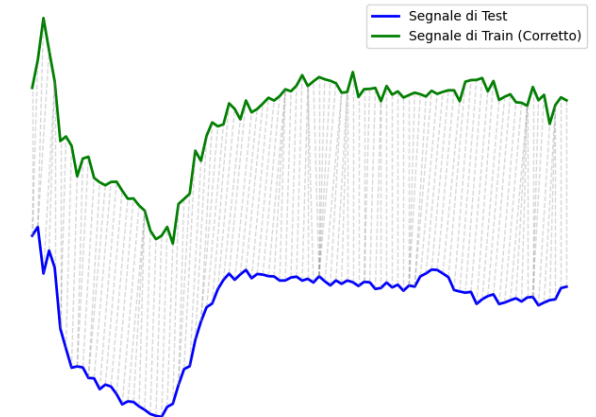
Osserviamo due casi nel dataset ECG200 in cui $W = 20\%$ porta ad errori di classificazione mentre $W = 4\%$ la classificazione avviene correttamente.



ECG200 sample 17



ECG200 sample 18

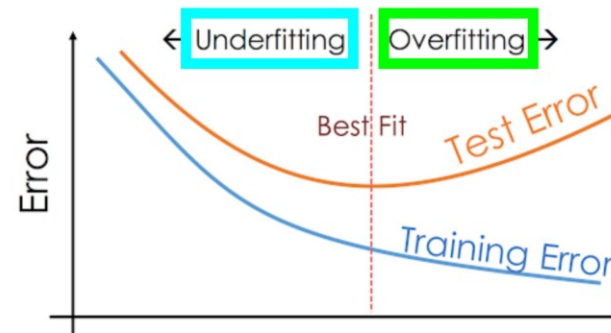


Risulta dunque evidente che al fine di un funzionamento ottimale di DTW all'interno dei problemi di classificazione sia necessario uno studio per definire per che valore di W si trovano i risultati migliori desiderati.

Ci stiamo muovendo all'interno dello spazio dei problemi di Machine Learning dove da un lato abbiamo le scelte che porteranno a una minore variabilità e quindi spesso a Underfitting (nel nostro caso la Distanza Euclidea) e nell'altra direzione abbiamo l'Overfitting creato dal fenomeno di OverWarping.

Osserviamo inoltre che limitare le dimensioni di W non ci serve solo ad evitare i casi di OverWarping, ma è sempre necessario anche in casi come Trace dove la performance di DTW si "saturava"; anche in quei casi è fondamentale avere chiaro il valore minimo di W per non aumentare la complessità di calcolo dell'algoritmo inutilmente senza migliorare la performance fornita.

O comunque, è importante valutare se l'aumento di performance dato da una maggiore W sia sufficiente da giustificare il conseguente aumento in termini di calcolo.



-
- [1] M. Chen, Oracle, "Cos'è il machine learning?", *Oracle Italia*.
<https://www.oracle.com/it/artificial-intelligence/machine-learning/what-is-machine-learning/>.
- [2] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43-49, Feb. 1978.
- [3] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD Workshop*, vol. 10, no. 16, pp. 359-370, Seattle, WA, Jul. 1994.
- [4] H. A. Dau, E. Keogh, "The UCR Time Series Classification Archive", *University of California, Riverside*.
https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (Ultimo accesso: 23 agosto 2026).
- [5] C. A. Ratanamahatana e E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM)*, 2005, pp. 506-510.