Wrocław University of Science and Technology
Faculty of Fundamental Problems of Technology

Field: **Biomedical Engineering**
Specialization: **Medical Informatics**

ENGINEERING THESIS

Comparative analysis of algorithms for signal alignment between sequences considering outliers

Jakub Bogacz

Supervisor
Ph.D. Eng. Agnieszka Uryga

WROCŁAW, 2024

Abstract

The alignment of biomedical signals is crucial for understanding complex physiological relationships and enhancing diagnostic capabilities. This thesis presents a comparative analysis of alignment algorithms, focusing on Dynamic Time Warping (DTW) and its variant, Drop-DTW, in the context of cerebral autoregulation and the autonomic nervous system. The study evaluates these algorithms' performance in handling noise and outliers, typical challenges in biomedical signal processing. Data were collected from healthy volunteers, preprocessed using Empirical Mode Decomposition, and analyzed for alignment effectiveness through quantitative and qualitative metrics. Results indicate that Drop-DTW demonstrates superior robustness in managing outliers while preserving essential signal features. The findings contribute to advancing methodologies for signal alignment and support the development of tools for personalized medicine and neuroscience research.

Keywords: Dynamic Time Warping, Drop-DTW, Signal Alignment, Biomedical Signals, Cerebral Autoregulation, Autonomic Nervous System, Noise Handling, Outlier Management

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation

Clinical impact

The development and evaluation of algorithms for signal alignment have a significant role in the analysis of dynamic changes in physiological signals. These algorithms facilitate the study of critical processes and gives us insight into complicated functional relationships in medical data. Reliable signal alignment is essential for extracting meaningful insights from biomedical data, which, in turn, can support individualized patient care and enhance clinical decision-making. Aligning algorithms are crucial in applications like patient monitoring, where understanding the temporal dynamics of health data can inform risk stratification and treatment decisions [8].

Moreover, it is observed that the topic of aligning medical signals is emerging in many various ways in interdisciplinary clinical studies. An example of recent usage of standard Dynamic Time Warping (DTW) for physiological processes is shown in a study by Chen et al., 2022 [2], where the algorithm was applied for quasi-synchronous alignment of seismocardiography and echocardiogram signals. This method reduced errors in detecting key points in seismocardiography, improving the calculation of important diagnostic measures like the ratio of the pre-ejection period to the left ventricular ejection time. A different example is study by van der Does et al., 2023 [27], where the authors applied DTW to model dynamic symptom networks in Ecological Momentary Assessment data, revealing the temporal and structural relationships between symptoms in individuals with mood disorders compared to healthy controls.

Tackling practical signal processing issues

The inherent design of standard DTW requires that every point in one time series is matched with at least one point in another series, which can lead to distortions when outliers are present. Biomedical signals generally exhibit noise, outliers, and irregular patterns due to various factors such as sensor inaccuracies, instrumentation noise, or patient movement. Artifacts present in data can significantly affect the performance and accuracy of alignment, leading to incorrect interpretations of the underlying phenomena. This thesis aims to address these practical challenges by evaluating recently developed alignment algorithms that are more robust in the presence of outliers. The results can be applied to enhance the reliability of signal analysis in real-world biomedical applications and can be used in development of tools and methods for advancing research in neuroscience and physiological signal processing.

Contributing to the AUTOMATIC project

This thesis is directly connected to the ongoing SONATA 18 program by BrainLab (UMO-2022/47/D/ST7/00229) [18], which focuses on analyzing the relationship between the autonomic nervous system (ANS) and cerebral autoregulation (CA). As part of the AUTOMATIC project, the findings of this work will contribute to the broader goal of understanding the brain-hearth crosstalk. In addition, the pilot study by Uryga et al., 2023 [25] shows that the interrelationship between CA and the ANS is heterogeneous and varies from patient to patient in traumatic brain injury, which clearly suggests that there is potential to develop alignment-based predictive algorithms that can support personalized diagnostics.

Evaluating methods for cerebral autoregulation

Over the past three decades, various techniques have been developed to evaluate the quality of CA, and no single method has been universally accepted as a gold standard [3]. Given the concept that CA represents the dynamic relationship between blood pressure (stimulus or input) and cerebral blood flow (response or output), transfer function analysis has become the most widely used method in studies examining spontaneous blood pressure fluctuations. Nevertheless, as it is pointed out in review by Ronney et al., 2022 [20] in recommendation 21: "Time-domain methods should be considered as an alternative to transfer function analysis when multiple inputs are needed to model changes in cerebral blood flow." Given the non-stationary nature of CA and functional dependence on the ANS, based on quality and repeatability of alignment patterns, time warping algorithms can prove to be a useful method of obtaining reliable information about the chosen evaluation method.

1.2. Physiological Background

Autoregulation is a critical physiological mechanism that allows various systems within the human body to maintain stable blood flow despite fluctuations in systemic blood pressure. This phenomenon is particularly evident in the cerebral, coronary, and renal circulations, where autoregulation plays a vital role in ensuring adequate perfusion and metabolic function.

Cerebral autoregulation is perhaps the most extensively studied form of autoregulation. It refers to the brain's ability to maintain a relatively constant cerebral blood flow (CBF) across a range of mean arterial pressures, typically between 60 and 150 mmHg [13, 11]. This mechanism is crucial for protecting neural tissue from ischemia or hyperemia, which can lead to neurological deficits or damage [24]. Studies have shown that CA operates through complex myogenic, neurogenic, and metabolic pathways, allowing rapid adjustments in vascular resistance to maintain stable blood flow [23]. For instance, the dynamic nature of CA has been demonstrated to be effective even during physiological challenges such as spaceflight or hypoxia, where the brain must adapt to significant changes in environmental conditions [16].

Moreover, the interplay between different autoregulatory mechanisms across various organ systems highlights the integrated nature of physiological responses in the human body. For example, while CA is mainly influenced by local metabolic demands and systemic blood pressure, it can also be affected by ANS activity, which modulates vascular tone and blood flow distribution [17]. This interdependence suggests that autoregulation is not merely an isolated phenomenon, but rather part of a broader network of homeostatic mechanisms that work together to maintain physiological stability.

The dependence of CA on the ANS

The classic view of cerebrovascular physiology has been that blood flow and cerebral metabolism are tightly coupled under the influence of substances, such as H^+, adenosine, nitric oxide and K^+, which ensure a rapid and matched supply of blood. However, CA has been shown to be more complex and multifaceted. The concept of neural control, or neurogenically mediated regulation of cerebral blood flow, is a relatively recent development, despite the anatomical identification of such nerves likely dating back more than 500 years [10]. Current research indicates that sympathetic and cholinergic reflexes account for a substantial portion of the autoregulatory response, with sympathetic control being particularly dominant in the active region of autoregulation [14]. This suggests that the sympathetic nervous system is crucial for maintaining cerebral perfusion during periods of increased metabolic demand or systemic hypotension. The role of ANS in CA is vividly debated for several reasons, such as the complexity of ANS and cerebrovascular interactions, measurement flaws, differences in methods used to evaluate CBF with respect to ANS, and experimental strategies.

1.3. Technical Background

The development of time-series alignment algorithms has been driven by the need to compare and analyze sequences of data that exhibit variability in their temporal dynamics. These methods are essential in fields like speech recognition, signal processing, and biomedical analysis, where data is often subject to time distortions, noise, and irregular patterns. Early approaches to aligning sequences relied on simple statistical measures such as cross-correlation, which identifies time lags between two signals by maximizing their similarity. While effective for linear and stationary signals, cross-correlation fails to account for non-linear distortions, limiting its utility in more complex datasets. The introduction of dynamic programming techniques revolutionized sequence alignment by allowing for non-linear mapping between sequences. Among these, DTW, first formalized by Sakoe and Chiba, 1978 [22], provided a framework for aligning sequences with varying lengths and temporal shifts.

Dynamic Time Warping

Mathematical notation used to define algorithms and figures representing cost matrices are matched with ones in the article by Nikita et al., 2021 [5], as Drop-DTW algorithm (proposed in a paper and discussed in more detail below) is a core part of conducted analysis. Optimal alignment between two sequences is achieved under specific constrains. Let $X = \left[x_1, \ldots, x_N\right]^\top \in \mathbb{R}^{N\times d}$ and $Z = \left[z_1, \ldots, z_K\right]^\top \in \mathbb{R}^{K\times d}$ be the input sequences, where N and M are the lengths of the two input sequences and d is the dimensionality of each element. As the scope of this study focuses on comparing time series of one-dimensional derived parameters, the value of $d = 1$. The valid alignments between sequences are defined as a binary matrix $M \in \{0,1\}^{K\times N}$, of the pairwise correspondences, where $M_{i,j} = 1$ if z_i is matched to x_j, and 0 otherwise. Matching an element has a cost $C_{i,j}$ which is a measure of dissimilarity between elements and is defined in equations (1.2) and (1.4). DTW finds the alignment M^* between sequences Z and X that minimizes aggregate cost:

$$M^* = \arg\min_{M\in\mathcal{M}} \langle M, C\rangle = \arg\min_{M\in\mathcal{M}} \sum_{i,j} M_{i,j}C_{i,j}, \tag{1.1}$$

where $\langle M, C\rangle$ is the Frobenius inner product and $\mathcal{M}$ is the set of all feasible alignments that satisfy the constraints of monotonicity, continuity and matching endpoints ($M_{1,1} = M_{K,N} =$

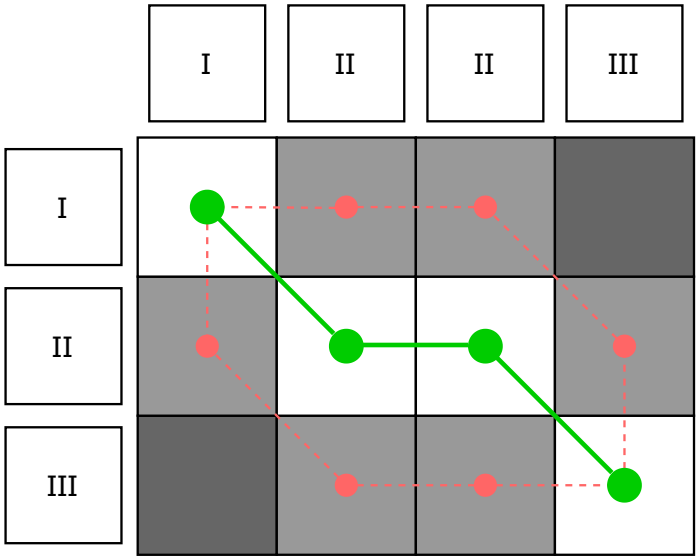


Figure 1.1: Visualization of the cost matrix used in the standard DTW algorithm

$1, \forall M \in \mathcal{M}$). The standard algorithm has a quadratic time and space complexity $O(NK)$. Nevertheless, further optimizations can be applied. Hirschberg's algorithm can reduce space consumption to $O(min(N, K))$ implementing a divide-and-conquer approach, splitting the problem into smaller subproblems recursively. The 50-year-old quadratic time bound has been overcome by Gold et al., 2018 [12], by taking a deterministic programming approach and thus reducing time complexity to $O(N^2/\log\log N)$.

The dynamic approach of this implementation can be visualized as in Figure 1.1. The colored matrices represent the pairwise matching costs $C_{i,j}$, with darker cell indicating higher cost. The paths on the grid are alignment paths, while the points on them indicate a pairwise match between the corresponding row and column elements. All three paths are feasible DTW paths, while only the green one indicates the optimal alignment with minimal cumulative cost.

Definition of match cost

The pairwise cost $C_{i,j}$ is defined based on distance between elements z_i and x_j. The implementation used in this work assumes that dropping elements from both Z and X is permitted, thus match cost is defined as:

$$C_{i,j} = 1 - \cos(z_i, x_j), \tag{1.2}$$

where $\cos(\theta)$ refers to the cosine similarity:

$$\cos(\theta) = \frac{z_i \cdot x_j}{\|z_i\|\|x_j\|}. \tag{1.3}$$

This method is the default approach for most machine learning oriented task, as it is perfect for measuring similarity of vectors in multidimensional space embeddings. For simpler tasks where z_i and x_j are respective scalar values, Euclidean distance can be used:

$$C_{i,j} = \sqrt{\sum_{k=1}^{n}(z_i - x_j)^2}. \tag{1.4}$$

Definition of drop cost

In order to handle outliers, there is a need to define dropping threshold values d^x and d^y. There are multiple methods to do that. The simplest approach is setting a constant s fixed across all elements in X and Z:

$$d_j^x = d_i^z = s. \tag{1.5}$$

It is important to note, however, that this approach poses considerable difficulties. Setting the constant s too low relative to the match costs $C_{i,j}$ can result in frequent drops, leading to the alignment of only a small portion of the signal. Conversely, if s is set too high, it may fail to exclude outliers, effectively preventing drops. To address that problem, s is defined as the $p \in [0, 100]$ top percentile of the values contained in the cost matrix C:

$$s = \text{percentile}\left(\{C_{i,j} \mid 1 \leq i \leq K, 1 \leq j \leq N\}, p\right). \tag{1.6}$$

Defining the drop cost as a function of the top percentile match costs is an intuitive way to apply a priori assumptions about the outlier rate in input sequences. If the expected noise level is unknown or changes between recordings, determining a fixed cost can be challenging. To address this issue, a learnable cost approach was adopted. The drop cost is evaluated based on the content of the corresponding sequence, allowing the cost to be dynamically adjusted to changing conditions. This approach is implemented as follows:

$$d^x = X \cdot f_{\omega_x}(\bar{Z}); \quad d^z = Z \cdot f_{\omega_z}(\bar{X}), \tag{1.7}$$

where $\bar{X}, \bar{Z}$ are means of sequences, and $f_\omega(\cdot)$ is a learnable function, i.e., feed forward neural net parametrized by ω. This definition can give rise to a more flexible system.

Drop-DTW

Drop-DTW extends consideration of feasible alignments $\mathcal{M}$ to those adhering to the monotonicity constraint only. Consequently, elements can be removed from the alignment process. In classical DTW, a naive approach to address outliers involves a two-step process: detect and drop outliers first, then align the remaining elements using DTW. However, this sequential method is suboptimal due to its inability to recover erroneously dropped elements and the risk of order-agnostic drops being spread across the entire sequence. Drop-DTW overcomes these issues by integrating outlier detection and sequence alignment into a unified framework that solves for the optimal temporal alignment while jointly enabling element dropping by adding another dimension to the dynamic programming table [5]. Specifically, dropping an element x_j indicates that no element from Z is associated with x_j. This is represented by the j-th column of the correspondence matrix M, which consists entirely of zeros, i.e. $M_{:,j} = 0$, same applies for $M_{:,i}$ when we consider matching z_i to X. For drop costs $d_i^z \in \mathbb{R}^K$ and $d_j^x \in \mathbb{R}^N$, for elements $z_i \in Z$ and $x_j \in X$, the optimal matching can be defined as:

$$M^* = \arg\min_{M \in \bar{\mathcal{M}}} \langle M, C \rangle + P_z(M) \cdot (d_i^z) + P_x(M) \cdot \left(d_j^x\right), \tag{1.8}$$

where $\bar{\mathcal{M}}$ is the set of binary matrices satisfying just the monotonicity constraint, and $P_x(M)$ is a vector with the j-th element equal to one if $M_{:,j} = 0$ and zero otherwise; $P_z(M)$ is defined similarly, but on rows.

Drop-DTW performs the alignment while considering dropping constraints, which slightly increases computational overhead, but does not change the asymptotic time complexity compared to DTW. The additional logic for handling outliers adds constant-time and space operations, which do not affect overall $O(NK)$ complexity.

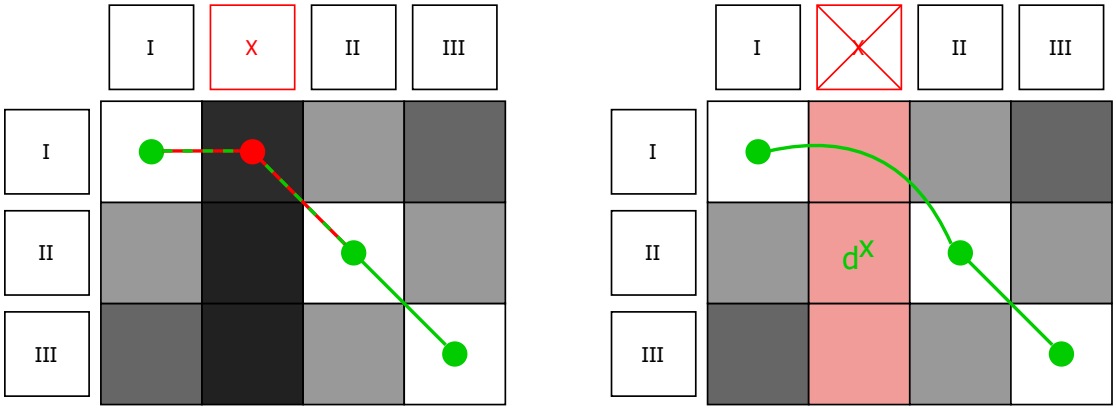


Figure 1.2: Comparison of cost matrices in presence of outlier: Classic DTW vs. Drop-DTW

The whole process can be represented as in Figure 1.2, where sequence X contains an outlier x_2. While classical approach have to incur a high cost (red point), Drop-DTW skips the outlier by paying the cost d_2^x and keeps only relevant matches.

Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) is defined as a process that decomposes a dataset into components known as Intrinsic Mode Functions (IMFs), satisfying specific criteria such as the equality of extreme points and zero crossings, and a zero mean of local envelopes, which therefore represents the oscillation mode embedded in the data. This method separates high-frequency components from baseline low-frequency trend lines from the input signal. The sifting process — iterative algorithm used to extract IMFs is defined as follows [15]:

$$X = \sum_{i=1}^{N} c_i + r_N, \tag{1.9}$$

where c_i are the IMFs and r_N is the residual, i.e., typically monotonic or trend-like signal that contains no oscillatory components and represents the low-frequency trend.

1.4. Hypotheses

This research is guided by two primary hypotheses:

1. **The Drop-DTW algorithm provides a more effective means than classic DTW for analyzing the relationship between CA and ANS.**
   It is hypothesized that the Drop-DTW algorithm provides a more effective approach than classical DTW when analyzing the relationship between CA and the ANS. Specifically, the ability of Drop-DTW to handle noisy data and outliers may result in superior alignment accuracy, thereby improving the quality of physiological signal analysis.
2. **The DropDTW algorithm performs significantly better in the presence of outliers than classical DTW.**
   The classical DTW algorithm differs significantly from the Drop-DTW algorithm, which can be observed through differences in processing outliers and noise, as well as through better signal alignment.

Chapter 2

Methodology

2.1. Data Acquisition

The data analyzed in this study were collected prospectively, as the planned analysis aimed to investigate dependence of cerebral autoregulation on the central nervous system. The study received ethical approval under the bioethics committee at Wroclaw Medical University, *no. KB-179/2023/N*, ensuring compliance with ethical standards. Data was gathered for the study conducted by Uryga et al., 2024 [26]. The dataset includes 37 healthy volunteers (21 females, 16 males, median age 22 years, range 18 – 31 years) who were measured at the Neuroengineering Laboratory at Wroclaw University of Science and Technology between October 2023 and January 2024. All subjects were free from any medication. They were asked to refrain from alcohol or caffeine consumption at least 12 hours before the examination, as proposed by Eames et al., 2004 [7]. Studies were conducted at room temperature with external stimuli minimized and under the supervision of a physician. Volunteers provided written consent before participation.

Arterial blood pressure (ABP) was measured noninvasively by a servo-controlled plethysmograph (CNAP, CNSystems Medizintechnik GmbH, Graz, Austria & Finapres Nova, FMS Medical Systems). The cuff was placed on the middle finger of the left hand and held at the level of the heart. Cerebral blood flow velocity (CBFV) was measured in the middle cerebral artery using transcranial Doppler ultrasonography (EMS-9PB, Delica, Shenzhen, China). A 2 MHz probe attached to a head frame was placed over the temporal window and fixed at a constant angle to obtain the optimal signal. The recording was collected from the left or right middle cerebral artery, depending on empirically estimated signal quality. Due to the relative symmetry of these vessels in healthy individuals, the analysis performed does not take into account from which side the recording was collected.

Due to the excessively high noise content of CBFV recordings, two measurements were discarded at the metrics determination stage. During analysis, one more was excluded due to missing values in critical metrics.

2.2. Derived Parameters Used in Analysis

CA was estimated using a mean flow velocity index (Mx) – moving correlation coefficient between slow-wave oscillations in ABP and CBFV within a 300-s moving time window advanced in 10-s steps. In this approach, it is assumed that over a short period of time autoregulation mechanism can be described by a linear dynamic system with mean arterial pressures as input and CBFV fluctuations as output [4]. It is worth noting that the optimal method for deriving Mx is currently unknown. It also remains to be established to which extent it provides meaningful clinical information in various clinical populations and healthy volunteers [19].

Heart rate (HR) – was estimated using Fast Fourier Transform (FFT) as the frequency of fundamental components of ABP in the frequency range, corresponding to heart cycle (40–140 beats per min; 0.67–2.33 Hz).

2.3. Computational Tools

This study utilized publicly available code to implement the algorithms under comparison. Specifically, Drop-DTW, developed by Nikita et al., 2021 [6], was edited and used under the terms of the Creative Commons License (CC BY-NC-SA 4.0). Additionally, the implementation of DTW and several visualization methods, including "two-way", "three-way", and "density" plot types, were provided by the dtw-python package [9]. While the original logic of the package was retained, minor modifications were introduced to its source code to accommodate the specific requirements of this analysis. The dtw-python package was selected for its reliability and versatility in handling diverse temporal signal types. Its comprehensive set of features, like flexibility, advanced visualization options and well written documentation, provided a solid foundation for the alignment tasks conducted in this research. To preprocess the data and extract features, Empirical Mode Decomposition was employed using the EMD Package [21]. Its versatility and adaptability to diverse signal characteristics make it a valuable computational resource in this study.

A complete set of code used in the analysis is provided in the attached archive. This includes all necessary scripts for data preprocessing, feature extraction, signal alignment using various algorithms (such as DTW and Drop-DTW), statistical analysis, and plotting. The code is organized and structured to ensure reproducibility of the results presented in the thesis.

2.4. Evaluation Metrics

Quantitatively, DTW provides a dissimilarity measure that can be employed to evaluate the similarity between time series. This measure is often compared to traditional metrics such as Euclidean distance, which has been shown to be less effective in the presence of temporal distortions [1]. The normalized dissimilarity measure, $\mathcal{C} = \sum C_{i,j}/(N + M)$, is defined as the sum of match costs of optimal path defined in Equation 1.1, divided by aggregate series lengths. For Drop-DTW, $\mathcal{C}$ is defined as the sum of both match and drop costs paid for the optimal path, divided by lengths. It has not been mathematically shown that both algorithms have fully comparable methods, as calculation of drop cost defined in equations from 1.5 to 1.7 have approach of different nature.

Qualitatively, the alignment produced by DTW and its variants can be visually inspected. This study employs three primary methods for visualizing alignment results: *two-way*, which presents a point-by-point comparison with matching lines, *three-way*, which facilitates a side-by-side inspection of the time series and their warping curve and *density*, which illustrates the cumulative cost landscape with the warping path superimposed. Evaluating the quality of time series alignment can be enabled through key features observable in plots:

- **Smooth Warping Path:** Gradual and consistent warping paths indicate effective alignment, avoiding abrupt or irregular jumps that suggest overfitting.
- **Minimal Crossing of Matching Lines:** In point-to-point comparison plots, fewer crossings between matching lines reflect better temporal correspondence.
- **Correspondence of Key Features:** Peaks, troughs, and other distinctive patterns should align accurately, highlighting meaningful relationships between sequences.

- **Cost Landscape Alignment:** Warping paths should traverse low-cost regions in cumulative cost density plots, avoiding suboptimal alignments in high-cost areas.

Combining visual inspection with quantitative metrics such as alignment cost or dropped points enhances the robustness of the evaluation. Testing on data with known correspondences provides a benchmark for alignment quality.

Chapter 3

Results

3.1. Data Preprocessing

In the process of selecting an appropriate pipeline for aligning the analyzed signals, several preprocessing techniques were considered, including Principal Component Analysis or application of simple filters. However, for the purposes of this study, the most effective results were obtained using EMD.

As illustrated in Figure 3.1, EMD has the notable advantage of preserving the characteristics of the input signal, as the Raw Signal exhibits the same overall structure as its reconstructed representation, obtained by summing the individual Intrinsic Mode Functions (IMFs). Furthermore, decomposing the signal into IMFs provides distinct components, each with its own unique characteristics. This decomposition enables selective control over the subsequent analysis by allowing the inclusion or exclusion of specific IMFs. For the purpose of this study, an author decided to use the last three modes of decomposed Mx signal, also as for HR, first mode was excluded. As low-order IMFs predominantly contain noise, they are excluded from further processing. Reconstructing input as $X' = c_{N-2} + c_{N-1} + r_N$ and $Z' = Z - c_1$ (See Equation 1.9.) is motivated by intrinsic properties of chosen modes. High-order IMFs, particularly the last few

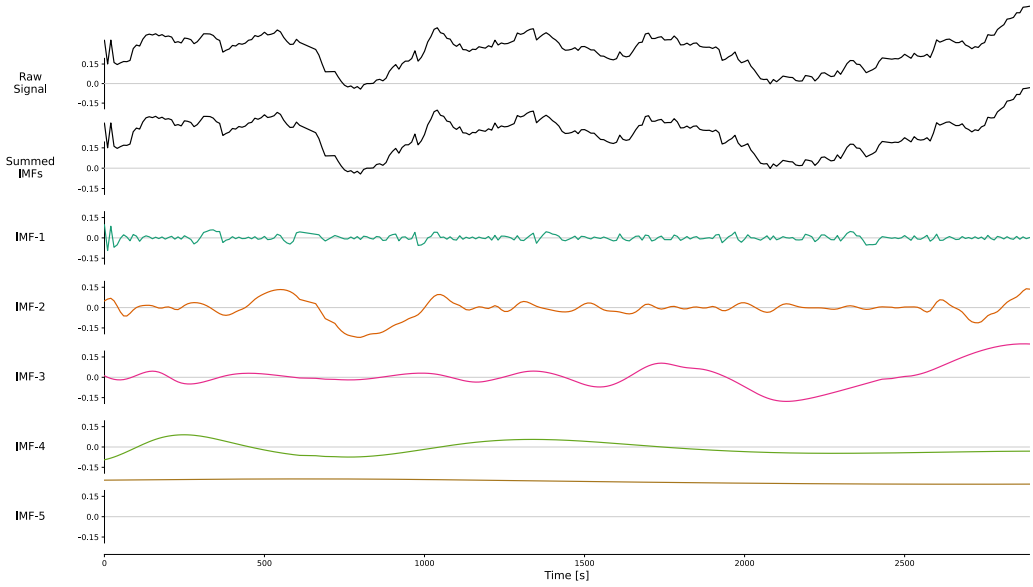
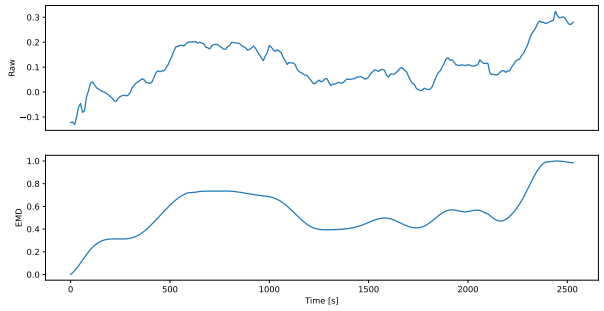
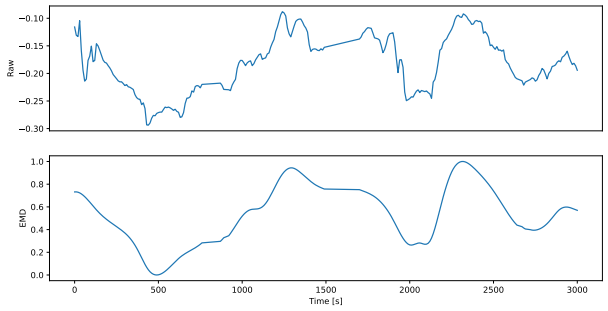


Figure 3.1: Empirical Mode Decomposition of mean flow velocity index series for selected subject

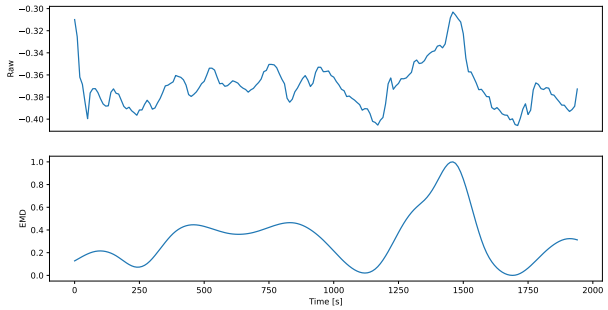
components, effectively captures the non-stationary trend, while retaining the most prominent oscillatory behaviors of the signal. This approach ensures a balance between removing noise and preserving the critical features necessary for downstream alignment, and emphasizes focus on slow-wave characteristics of analyzed data.



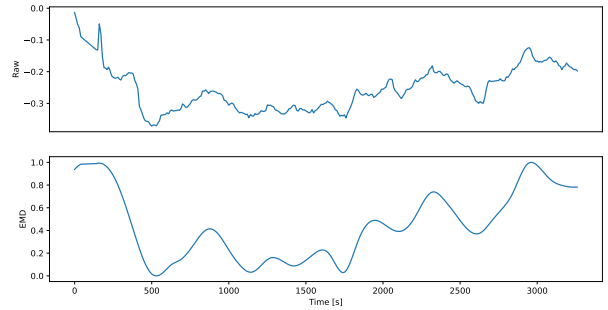
(a) Recording A



(b) Recording B

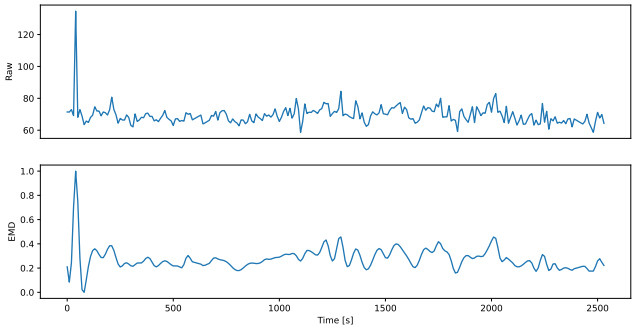


(c) Recording C

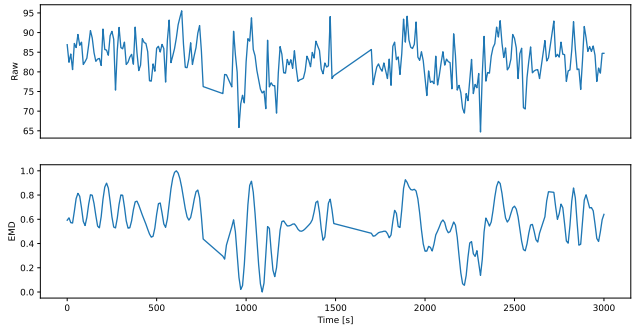


(d) Recording D

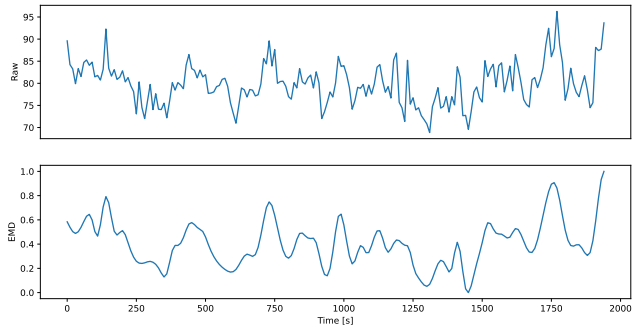
Figure 3.2: Comparison of **raw and preprocessed Mx** signals for selected subjects



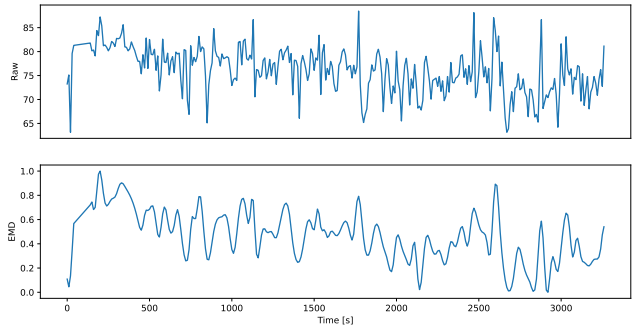
(a) Recording A



(b) Recording B



(c) Recording C



(d) Recording D

Figure 3.3: Comparison of **raw and preprocessed HR** signals for selected subjects

3.2. Analysis of Alignment Costs

The minimal aggregate match costs of the optimal alignment paths were calculated for all 34 recordings. To ensure comparability across subjects of varying lengths, these costs were normalized. The normalized distances for the DTW algorithm ranged from a minimum of 0.0457 (subject 33) to a maximum of 0.2102 (subject 12). For the DropDTW algorithm, the normalized distances ranged from a minimum of 0.0786 (subject 34) to a maximum of 0.2050 (subject 12). Introducing the 75th top percentile drop cost, as defined in Equation 1.6, resulted in an average increase of normalized aggregate cost values by 0.0422.

Table 3.1: Summary of minimal aggregate alignment costs for DTW and DropDTW algorithms performed on Mx and HR series. The normalization accounts for varying sequence lengths, ensuring comparability across subjects

Subject	DTW	DTW Normalized	DropDTW	DropDTW Normalized
1	32.5901	0.0853	56.4067	0.1477
2	43.3572	0.0771	78.5985	0.1399
3	83.7037	0.1453	102.1738	0.1774
4	29.9984	0.0735	44.7771	0.1097
5	29.4649	0.0649	55.0662	0.1213
6	30.0617	0.0737	48.1215	0.1179
7	31.7911	0.0600	63.4308	0.1197
8	35.1098	0.1211	38.9504	0.1343
9	32.6856	0.0641	64.8754	0.1272
10	40.8490	0.0714	69.7501	0.1219
11	49.8488	0.0834	88.0221	0.1472
12	105.0935	0.2102	102.5132	0.2050
13	34.9915	0.0566	55.9765	0.0906
14	32.5988	0.0665	46.6702	0.0952
15	61.6572	0.0969	101.7327	0.1600
16	12.5603	0.0465	22.7711	0.0843
17	14.5753	0.0548	21.9764	0.0826
18	41.2522	0.0618	70.8630	0.1061
19	14.9259	0.0786	20.2765	0.1067
20	55.8106	0.0921	61.5483	0.1016
21	53.3622	0.0794	91.3665	0.1360
22	44.1879	0.0658	79.6802	0.1186
23	50.4373	0.0721	79.8260	0.1140
24	47.0637	0.0964	76.0901	0.1559
25	37.9639	0.1157	49.7540	0.1517
26	35.9592	0.1016	52.5980	0.1486
27	36.6713	0.0687	60.2861	0.1129
28	36.8063	0.0579	67.2461	0.1057
29	35.1360	0.0665	65.7997	0.1246
30	36.9993	0.1022	51.7371	0.1429
31	42.2492	0.1089	56.1739	0.1448
32	34.9606	0.0896	45.2757	0.1161
33	17.5609	0.0457	37.8245	0.0985
34	40.4703	0.0559	56.8733	0.0786

Distribution testing of the DTW normalized distances:

- $p = 3.5617 \times 10^{-5}$ (Shapiro-Wilk test),
- $p = 1.4233 \times 10^{-7}$ (D'Agostino and Pearson's test).

The data does not follow a normal distribution.

Distribution testing of the DropDTW normalized distances:

- $p = 0.3818$ (Shapiro-Wilk test),
- $p = 0.1356$ (D'Agostino and Pearson's test).

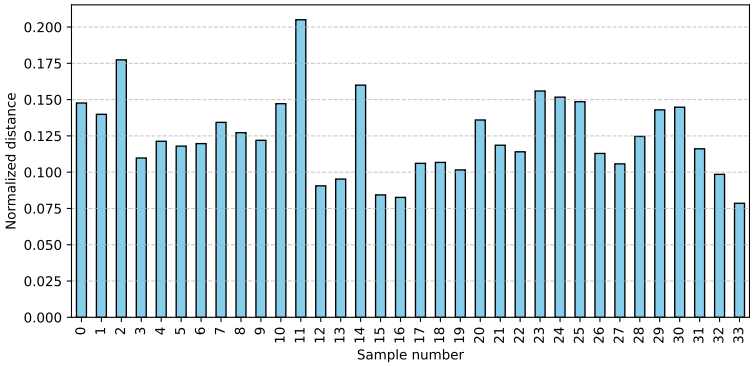
The data follows a normal distribution.

Population testing of the normalized distances:

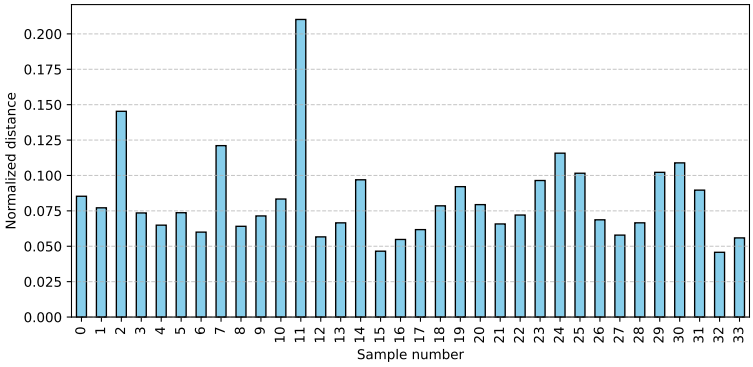
- $U = 138.0000$ (Mann-Whitney U Test),
- $p = 7.0180 \times 10^{-8}$ (Mann-Whitney U Test).

The distributions of DTW and DropDTW normalized distances are significantly different.

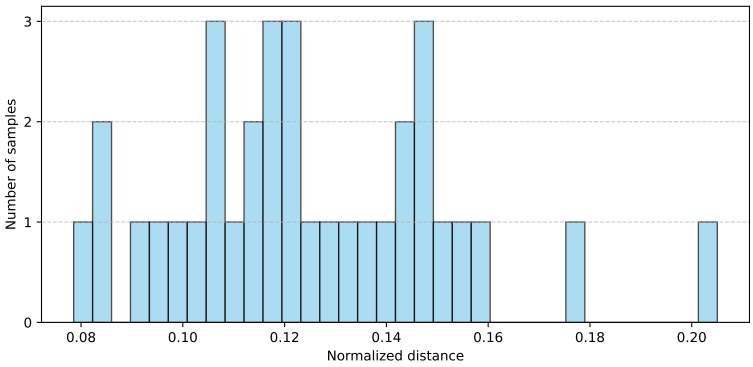
Alignments conducted with the classical DTW algorithm proved to be unreliable for signal pairs producing the highest alignment costs. Specifically, these alignments exhibited distorted patterns, such as many warping paths converging on a single point, excessive warping leaps, and inconsistent warping patterns. Conversely, for signal pairs with the lowest alignment costs, the author's implementation of DropDTW appears to apply excessive constraints, leading to suboptimal results. In these cases, DropDTW showed minimal warping and demonstrated little to no meaningful alignment, suggesting that the algorithm's robustness may unnecessarily suppress alignment flexibility when alignment costs are already minimal.



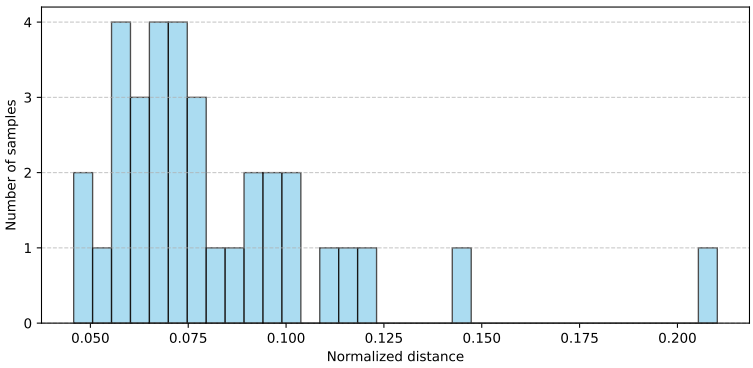
(a) Bar chart of DropDTW distances



(b) Bar chart of DTW distances



(c) Histogram of DropDTW distances



(d) Histogram of DTW distances

Figure 3.4: Comparison of DTW and Drop-DTW performance. The bar plots (a, b) illustrate distances for given subjects, while the histograms (c, d) depict the distribution of alignment costs

3.3. Observations on Alignment Behavior

A DTW algorithm exhibited sensitivity to abrupt peaks. These peaks, characterized by sudden increases in value, led to significantly high costs in the DTW cost matrix. This resulted in alignment distortions, with the algorithm aligning multiple points from one signal to the point in the other signal. This phenomenon was particularly evident in the "two-way" plots, where warping lines clustered around one or few samples, indicating a disproportionate allocation of alignment. In the "three-way" plots, this behavior manifested itself as sharp steps in the alignment curve. Instead of a smooth trajectory, indicative of balanced alignment, the steps highlight regions where the alignment cost abruptly changed. Such patterns suggest that DTW struggles to handle outliers effectively, as these regions dominate the alignment process and distort the overall result. For cases of effective alignment, both the "two-way" and "tree-way" plots exhibited consistent patterns. The "two-way" plots showed evenly distributed warping lines without crossings, while the "three-way" plots displayed a diagonal trajectory, indicative of a balanced and uniform alignment across the sequence. These observations highlight the dual nature of DTW's performance: its effectiveness in handling typical signal variations and its vulnerability to outlier-driven distortions.

Figures below provide an illustrative view of the alignment between min-max normalized Mx and HR series. In these plots, the green dashed line shows the mean of the normalized Mx series, while the red dashed line represents the mean of the normalized HR series. For the DropDTW plots, blue dots indicate dropped values from the Mx series and red dots represent dropped values from the HR series. Dropped values correspond to points that were identified by DropDTW as having an outsized impact on alignment costs, and are calculated using drop method based on the top 75-th percentile. It is important to note that the difference in density of warping lines observed between DTW and DropDTW plots does not reflect a meaningful difference in alignment behavior. This variation arises from differences in the implementations of plotting algorithms and does not alter the interpretation of the plots or the alignment results. Recordings A, B, C, and D are selected from different quarters of the cost distribution.

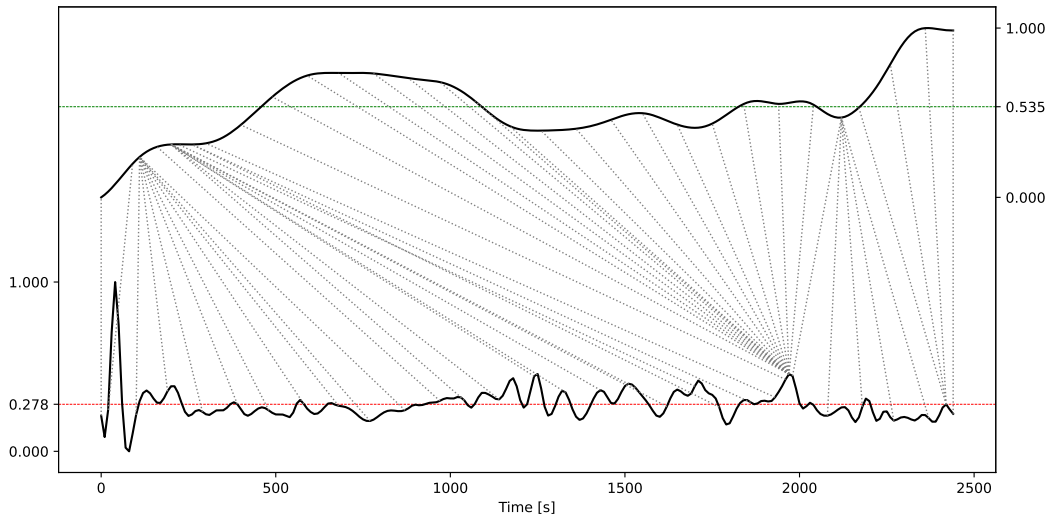


Figure 3.5: Two-way plot of **recording A** showcasing defective alignment with **DTW**. Numerous warping lines converge on a single point, indicating distortion caused by a sharp peak in the signal

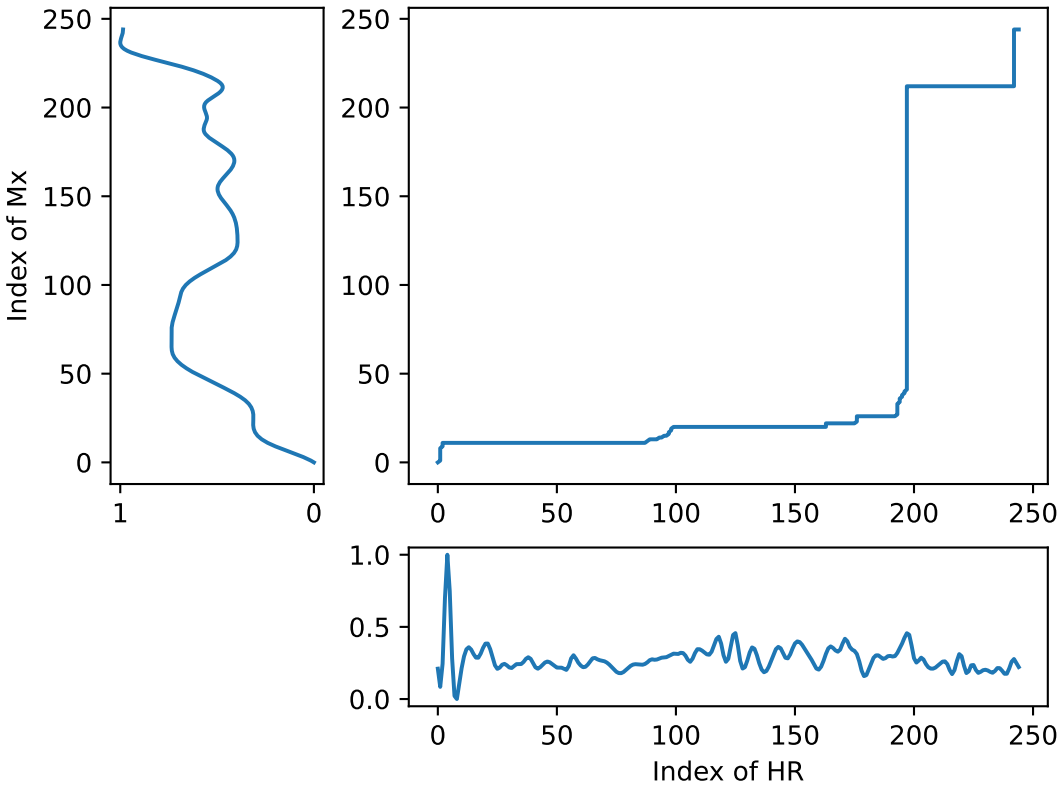


Figure 3.6: Three-way plot of **recording A** with defective alignment returned by **DTW**. The alignment curve displays a pronounced sharp step, reflecting a high-cost region

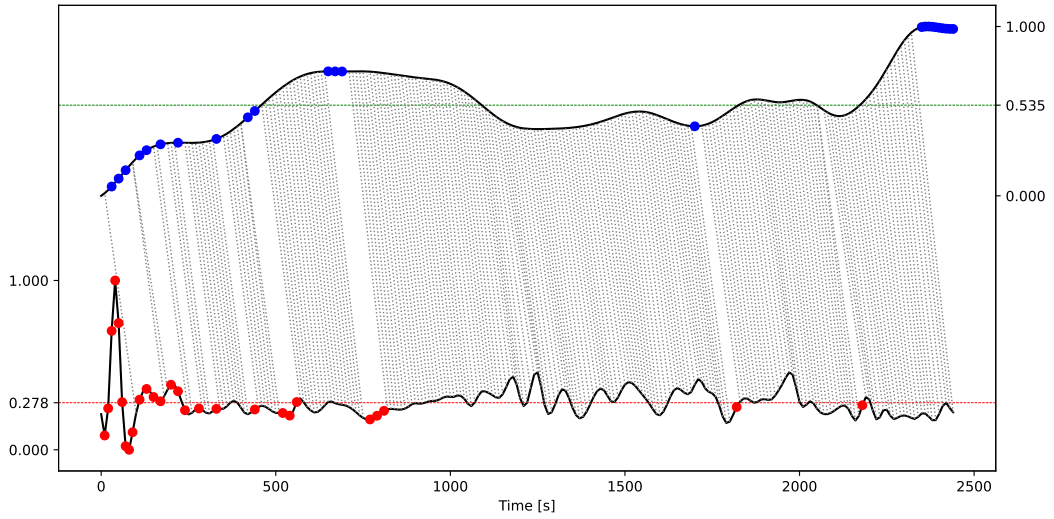


Figure 3.7: Two-way plot of **recording A** showcasing improved alignment with **DropDTW**. Drops in alignment cost are visible for the majority of abrupt values, with no evidence of multiple points from one sequence aligning disproportionately to a single point in the other sequence

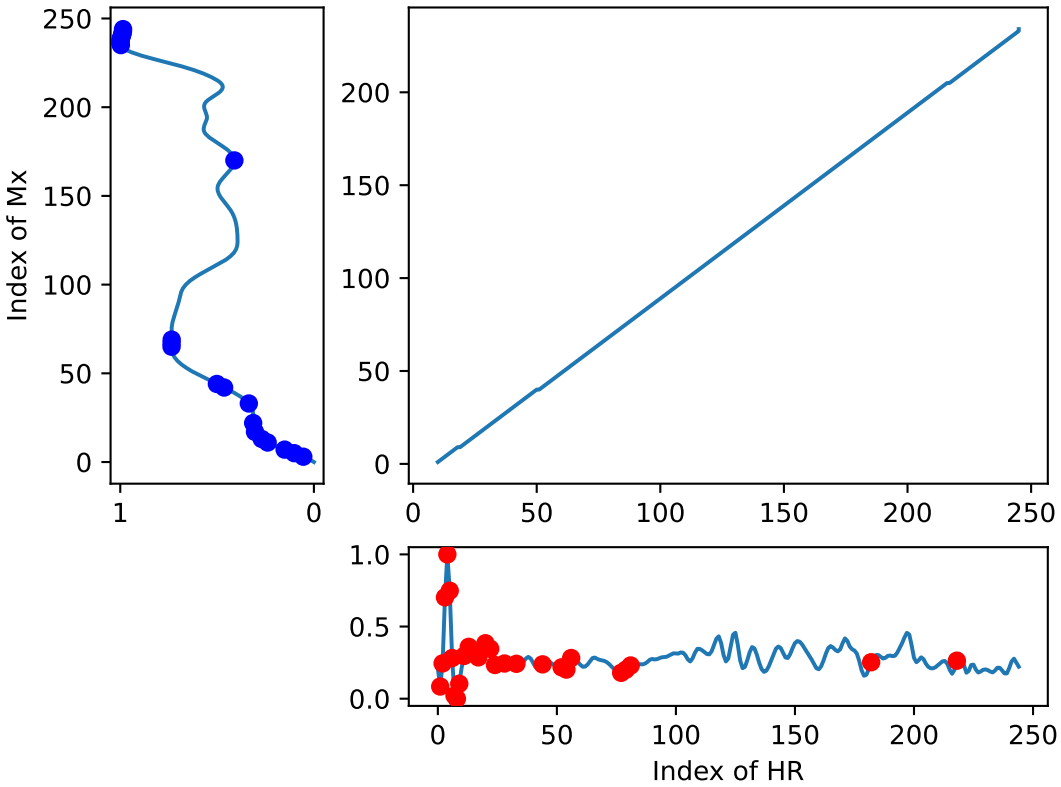


Figure 3.8: Three-way plot of **recording A** with **DropDTW** showing improved alignment. The sharp step, previously indicative of defective alignment, has disappeared, reflecting a more balanced alignment cost distribution

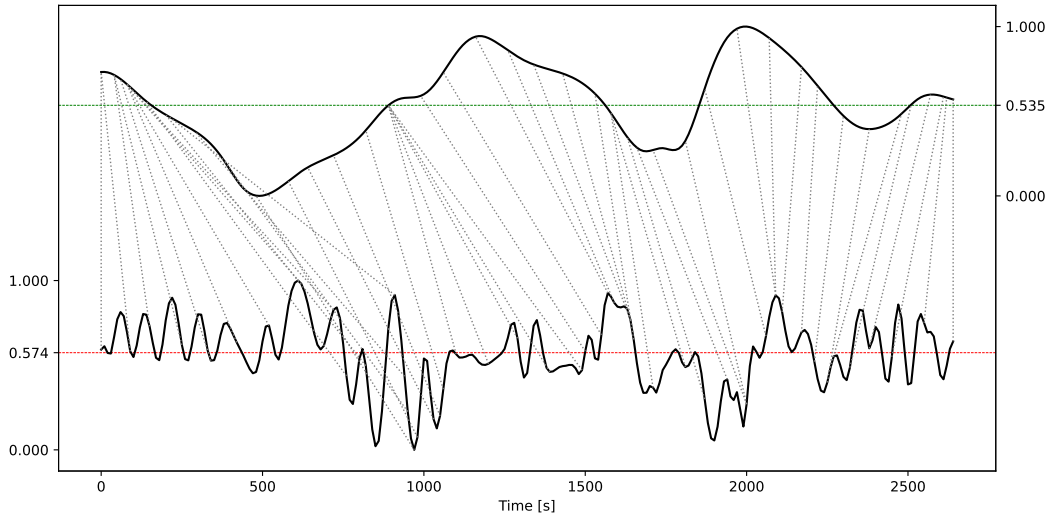


Figure 3.9: Two-way plot of **recording B** illustrating sequences well-aligned with **DTW**. Warping lines are evenly distributed, demonstrating a balanced alignment across the sequences

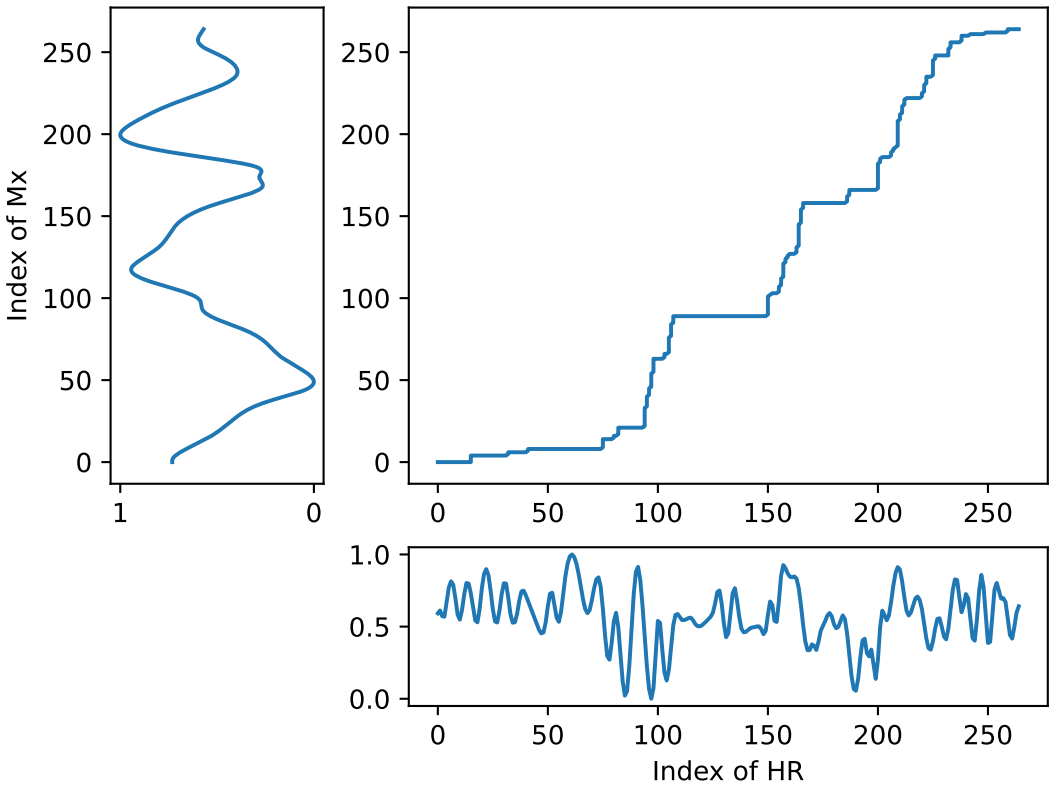


Figure 3.10: Three-way plot of **recording B** demonstrating effective alignment with **DTW**. The alignment curve follows a smooth diagonal trajectory, indicative of uniform alignment over the sequence

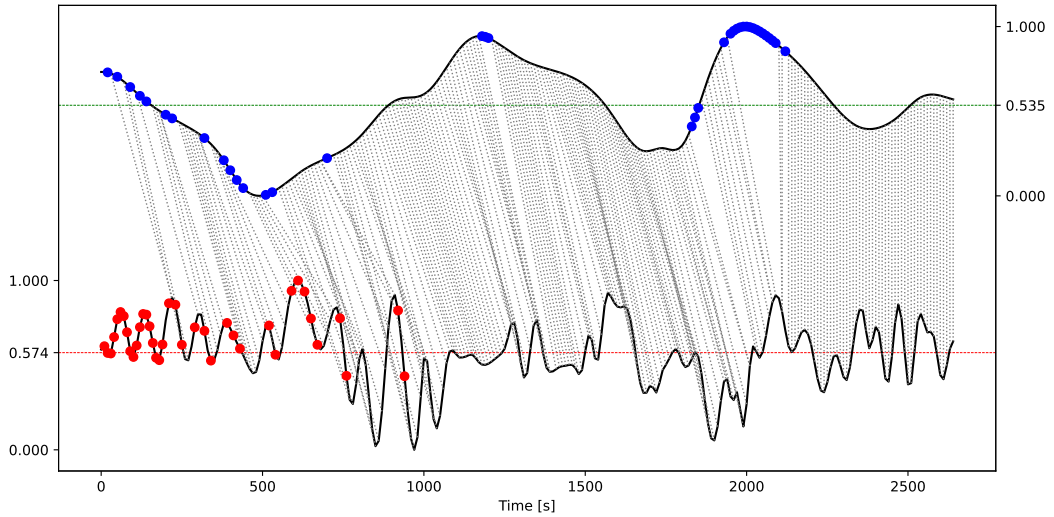


Figure 3.11: Two-way plot of **recording B** illustrating alignment with **DropDTW**. The alignment process warps over shorter time periods compared to the standard DTW algorithm, reflecting a tendency for localized alignment

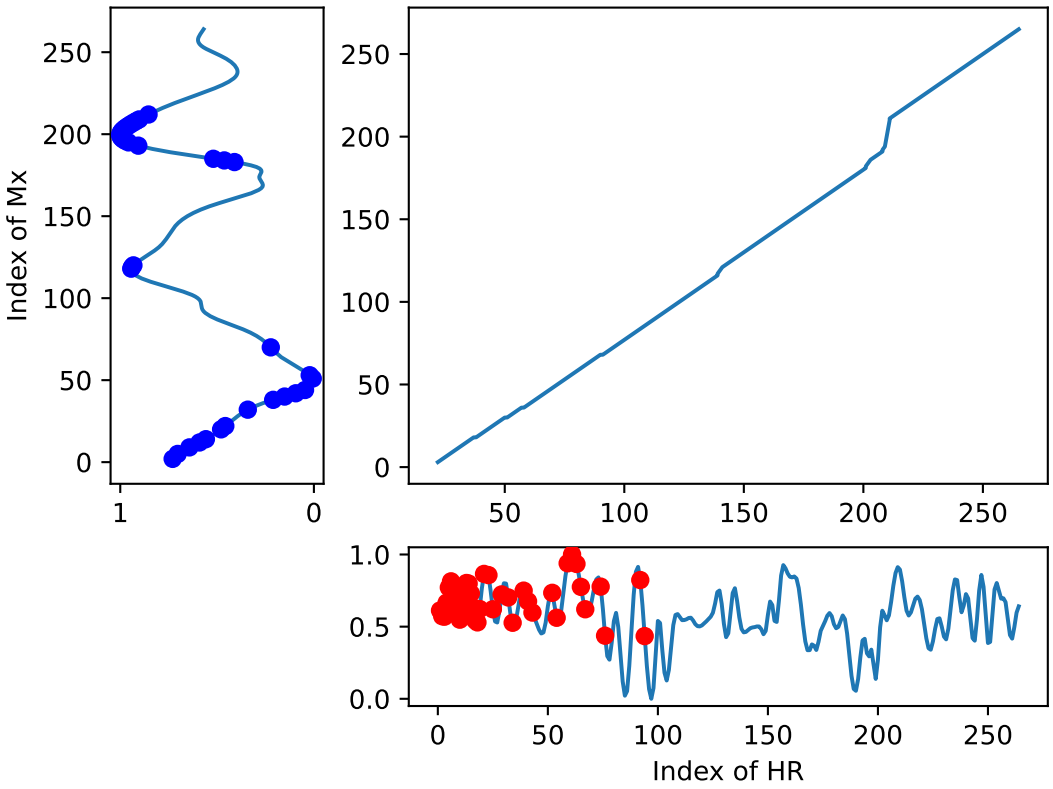


Figure 3.12: Three-way plot of **recording B** demonstrating alignment using **DropDTW**. The alignment curve is almost fully diagonal, indicative of a uniform and consistent alignment across the sequences

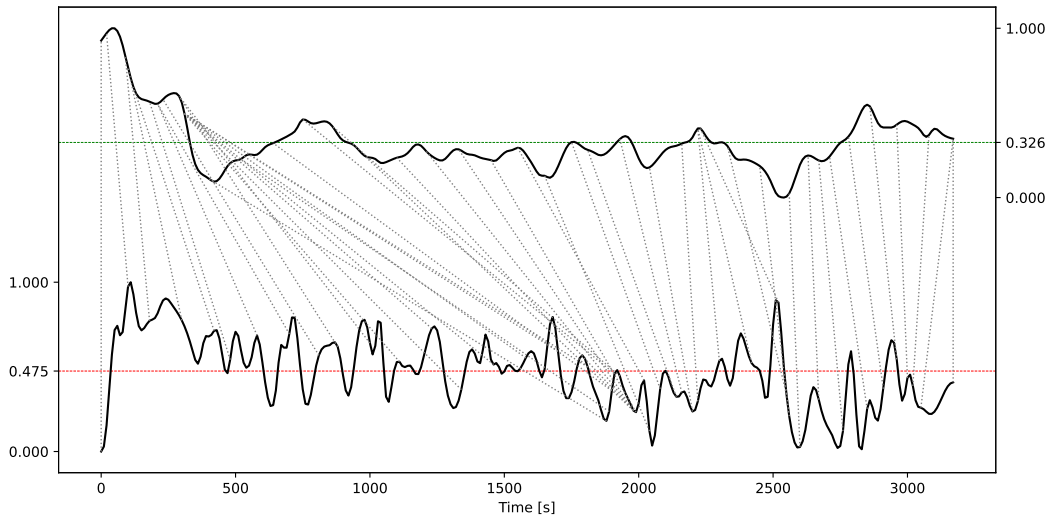


Figure 3.13: Two-way plot of **recording C** showcasing alignment with **DTW**

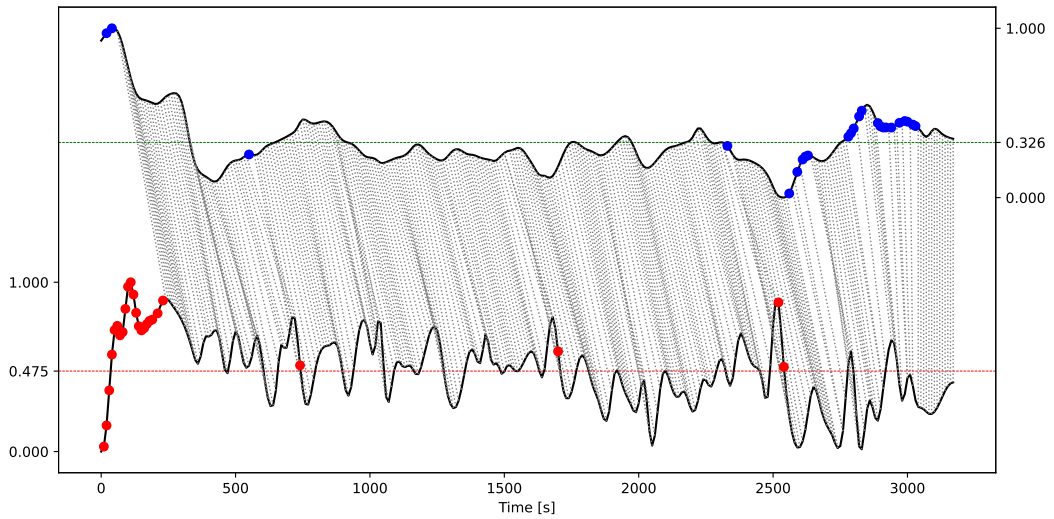


Figure 3.14: Two-way plot of **recording C** showcasing alignment with **DropDTW**

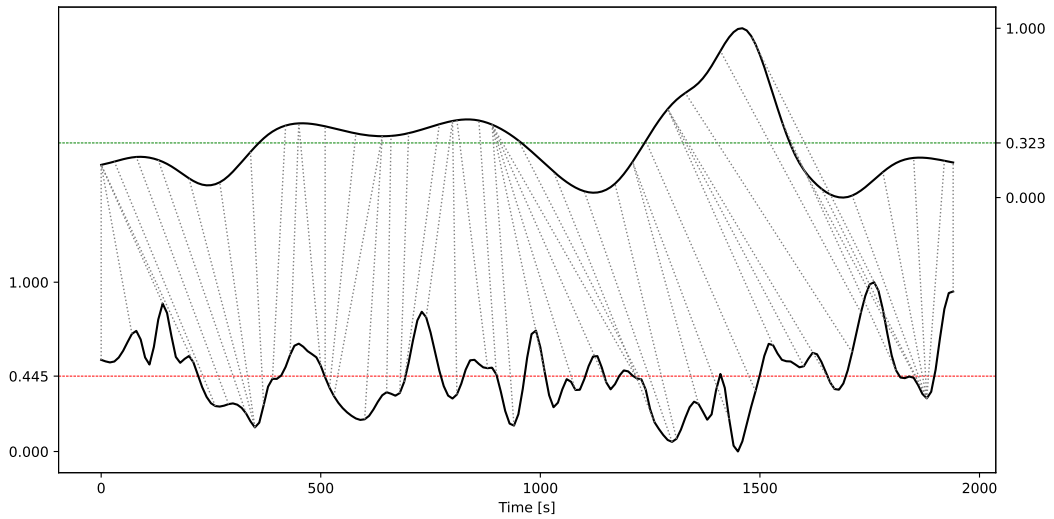


Figure 3.15: Two-way plot of **recording D** showcasing alignment with **DTW**

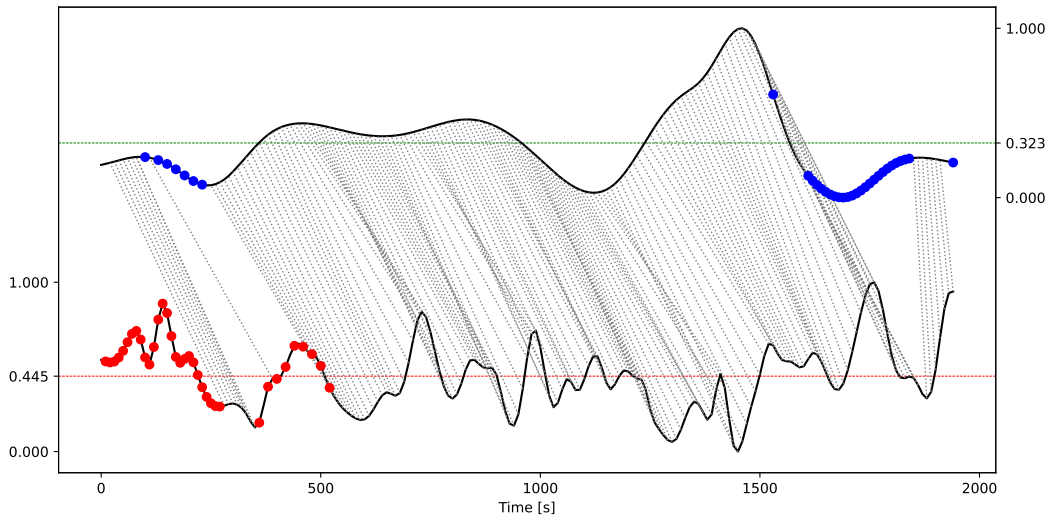


Figure 3.16: Two-way plot of **recording D** showcasing alignment with **DropDTW**

Chapter 4

Discussion

4.1. Key Findings

Both algorithms, DTW and Drop-DTW, exhibit the same computational complexity, as detailed in the Technical Background section. This shared characteristic poses challenges when handling large datasets, as the quadratic computational cost becomes a limiting factor.

Both DTW and DropDTW algorithms face distinct challenges that influence their practical utility in signal alignment tasks. Classical DTW, while widely used due to its simplicity and flexibility, struggles to handle signals with significant discrepancies or the presence of outliers. This issue is evident in alignments with high aggregate costs, where warping paths often converge on single points, exhibit large leaps, or display inconsistent patterns. DropDTW, on the other hand, addresses some of the limitations of classical DTW by selectively excluding high-cost regions. However, this approach introduces its own set of challenges. For signal pairs with small aggregate alignment costs, DropDTW demonstrates reduced warping ability, resulting in less meaningful alignment. This suggests that its stringent cost-reduction mechanism can become redundant or even counterproductive in cases where signals are already well-matched. While this behavior ensures computational efficiency and robustness against outliers, it limits DropDTW's adaptability in scenarios where subtle variations in the alignment could hold critical information.

Regarding the first hypothesis, **the Drop-DTW algorithm provides a more effective means than classic DTW for analyzing the relationship between CA and ANS**, the findings suggest that the differences between the algorithms are not exhaustive enough to draw definitive conclusions about their relative effectiveness in capturing the relationship between CA and the ANS. Inconsistencies in the results between subjects indicate that more data and research is needed to fully assess the potential of Drop-DTW in this context. Furthermore, limiting the analysis to a single set of patients does not allow for a comprehensive view of the alignment between normal and abnormal signals. This implies that further investigation, potentially incorporating data from patients with impaired CA, may be required to reveal the full capabilities of Drop-DTW to study CA and ANS.

Regarding the second hypothesis, **the DropDTW algorithm performs significantly better in the presence of outliers than the classical DTW**, the findings suggest that DropDTW offers distinct advantages over the classical DTW in handling datasets with significant outliers or noise. DropDTW's selective exclusion mechanism, which removes high-cost regions from the alignment process, demonstrates robustness against outlier-driven distortions. This is evident from its ability to maintain meaningful alignments despite the presence of abrupt peaks or discrepancies, as opposed to classical DTW, which struggles with alignment distortions under similar conditions.

4.2. Limitations of the Study

Despite the significant progress achieved in this study, several limitations must be acknowledged. These limitations highlight areas where future research could expand and improve upon the current findings.

1. **Unsuccessful implementation of learnable drop cost function**
   A key challenge encountered was the unsuccessful adaptation of the learnable drop cost function, as defined in Equation (1.7) and described by Nikita et al. [5]. Originally developed for video embeddings, this function proved difficult to integrate into algorithms tailored for time-series data. At the same time, the fixed percentile drop cost proved insufficiently flexible for the analyzed dataset, as outlier values varied significantly depending on the recording. Further exploration into modifying or redesigning such functions for time-series applications remains a promising avenue for future work.
2. **Insufficient signal length for physiological insights**
   The recordings used in this study were relatively short, which restricted the ability to draw broader conclusions about the physiological functional relationship between signals representing CA and the ANS. Dynamic time warping steps, crucial for assessing temporal alignment, require longer sequences to capture nuanced interactions and variability. As a result, the study's findings provide a limited perspective on these complex relationships. Future research would benefit from using longer recording durations to enhance the depth and reliability of physiological interpretations.
3. **Restricted dataset scope**
   A crucial limitation of the present study lies in its scope: all analyses were conducted on data from healthy control subjects. While this provides a controlled environment to evaluate algorithm performance, it restricts the broader applicability of the findings. Without extending the analysis to data from patients with medical conditions such as traumatic brain injury [25], these alignments remain theoretical constructs with limited real-world significance. Incorporating both healthy and pathological data in future studies could enhance the practical relevance of these algorithms and provide deeper insights for tuning their implementation. For instance, alignment results from diverse datasets could serve as input for machine learning models in representation learning, potentially uncovering meaningful patterns that could inform clinical diagnostics or other practical applications.

By addressing these limitations in future work, it will be possible to build on the foundation laid by this study and achieve more robust and comprehensive insights into signal alignment and its physiological implications.

4.3. Final Remarks

In this thesis, an author focused on simple methods for assessing the quality of sequence alignment: a quantitative analysis based on the cumulative cost of the best alignment and a qualitative analysis involving visual inspection of the results. While this approach provides valuable insights, it is constrained by the lack of advanced automated analytical methods.

Based on the available literature, e.g. [1], several more sophisticated methods can be identified that could be applied in future research. For instance:

- **Representation learning**: Transforming sequences into feature spaces using techniques such as Shapelet Transform or Embedded Features. This enables the use of vector-based classifiers such as Support Vector Machines (SVMs).

- **Kernel methods**: Developing kernel functions based on distance measures, such as Recursive Edit Distance Kernel (REDK) or Gaussian Distance Substitution Kernel, to exploit the strengths of kernel-based algorithms.
- **Dimensionality reduction and prototype selection**: Applying different techniques selecting representative prototypes to significantly reduce computational cost while preserving essential information.
- **Hybrid approaches**: Combining quantitative methods with machine learning techniques to better capture both global and local patterns.

Integrating these techniques in future studies could not only enhance the precision of the analysis but also broaden its applicability to more complex datasets and research scenarios.

Bibliography

[1] A. Abanda, U. Mori, J. Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33:378–412, 2018.

[2] C. Chen, W. Lin, M. Lee. Computer-aided detection of fiducial points in seismocardiography through dynamic time warping. *Biosensors*, 12:374, 2022.

[3] J. A. Claassen, A. S. Meel-van den Abeelen, D. M. Simpson, R. B. Panerai. Transfer function analysis of dynamic cerebral autoregulation: A white paper from the international cerebral autoregulation research network. *Journal of Cerebral Blood Flow & Metabolism*, 36(4):665–680, 2016.

[4] M. Czosnyka, K. Brady, M. Reinhard, P. Smielewski, L. A. Steiner. Monitoring of cerebrovascular autoregulation: Facts, myths, and missing links. *Neurocritical Care*, 10:373–386, 2009.

[5] N. Dvornik, I. Hadji, K. G. Derpanis, A. Garg, A. D. Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. *CoRR*, abs/2108.11996, 2021.

[6] N. Dvornik, I. Hadji, K. G. Derpanis, A. Garg, A. D. Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. `https://github.com/SamsungLabs/Drop-DTW`, 2023. Accessed: 2024-12-02.

[7] P. J. Eames, J. F. Potter, R. B. Panerai. Influence of controlled breathing patterns on cerebrovascular autoregulation and cardiac baroreceptor sensitivity. *Clinical Science*, 106:155–162, 2004.

[8] A. Giannoula, M. Gutiérrez, Bravo, F. Sanz, L. Furlong. Identifying temporal patterns in patient disease trajectories using dynamic time warping: a population-based study. *Scientific Reports*, 8, 2018.

[9] T. Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009.

[10] P. J. Goadsby. Chapter 16 - autonomic nervous system control of the cerebral circulation. R. M. Buijs, D. F. Swaab, redaktorzy, *Autonomic Nervous System*, wolumen 117 serii *Handbook of Clinical Neurology*, strony 193–201. Elsevier, 2013.

[11] N. Goettel, C. Patet, A. Rossi, C. Burkhart, M. Czosnyka, S. Strebel, L. Steiner. Monitoring of cerebral blood flow autoregulation in adults undergoing sevoflurane anesthesia: a prospective cohort study of two age groups. *Journal of Clinical Monitoring and Computing*, 30:255–264, 2015.

[12] O. Gold, M. Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4), Sier. 2018.

[13] W.-t. Guo, H. Ma, J. Liu, Z.-N. Guo, Y. Yang. Dynamic cerebral autoregulation remains stable during the daytime (8 a.m. to 8 p.m.) in healthy adults. *Frontiers in Physiology*, 9, 2018.

[14] J. Hamner, C. Tan. Relative contributions of sympathetic, cholinergic, and myogenic mechanisms to cerebral autoregulation. *Stroke*, 45:1771–1777, 2014.

[15] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, H. H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998.

[16] K. Iwasaki, B. Levine, R. Zhang, J. Zuckerman, J. Pawelczyk, A. Diedrich, A. Ertl, J. Cox, W. Cooke, C. Giller, C. Ray, L. Lane, J. Buckey, F. Baisch, D. Eckberg, D. Robertson, I. Biaggioni, C. Blomqvist. Human cerebral autoregulation before, during and after spaceflight. *The Journal of Physiology*, 579:799–810, 2007.

[17] B. Kolb, D. Rotella, H. Stauss. Frequency response characteristics of cerebral blood flow autoregulation in rats. *Ajp Heart and Circulatory Physiology*, 292:H432–H438, 2007.

[18] Narodowe Centrum Nauki. Nr rejestracyjny: 2022/47/D/ST7/00229, 2022. Accessed: Nov. 24, 2024.

[19] M. H. Olsen, C. G. Riberholt, J. Mehlsen, R. M. Berg, K. Møller. Reliability and validity of the mean flow index (mx) for assessing cerebral autoregulation in humans: A systematic review of the methodology. *Journal of Cerebral Blood Flow & Metabolism*, 42(1):27–38, 2022.

[20] R. B. Panerai, P. Brassard, J. S. Burma, P. Castro, J. A. Claassen, J. J. van Lieshout, J. Liu, S. J. Lucas, J. S. Minhas, G. D. Mitsis, R. C. Nogueira, S. Ogoh, S. J. Payne, C. A. Rickards, A. D. Robertson, G. D. Rodrigues, J. D. Smirl, D. M. Simpson, on behalf of the Cerebrovascular Research Network (CARNet). Transfer function analysis of dynamic cerebral autoregulation: A carnet white paper 2022 update. *Journal of Cerebral Blood Flow & Metabolism*, 43(1):3–25, 2023.

[21] A. J. Quinn, V. Lopes-dos Santos, D. Dupret, A. C. Nobre, M. W. Woolrich. Emd: Empirical mode decomposition and hilbert-huang spectral analyses in python. *Journal of Open Source Software*, 6(59):2977, 2021.

[22] H. Sakoe, S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[23] C. Tan, J. Hamner, J. Taylor. The role of myogenic mechanisms in human cerebrovascular regulation. *The Journal of Physiology*, 591:5095–5105, 2013.

[24] C. Tan, J. Taylor. Integrative physiological and computational approaches to understand autonomic control of cerebral autoregulation. *Experimental Physiology*, 99:3–15, 2013.

[25] A. Uryga, M. Kasprowicz, M. Burzyńska, A. Kazimierska, M. Czosnyka, N. Nasr. Association between temporal patterns of baroreflex sensitivity after traumatic brain injury and prognosis: a preliminary study. *Neurological Sciences: Official Journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology*, 44(5):1653–1663, 2023.

[26] A. Uryga, M. Najda, I. Berent, C. Mataczyński, P. Urbański, M. Kasprowicz, T. Buchner. The impact of controlled breathing on autonomic nervous system modulation: analysis

using phase-rectified signal averaging, entropy and heart rate variability. *Physiological Measurement*, 45(9):2–3, Sep 2024.

[27] F. van der Does, W. van Eeden, F. Lamers, B. Penninx, H. Riese, E. Vermetten, K. Wardenaar, N. van der Wee, E. Giltay. Big data networks: Dynamic time warping as a statistical tool for network analysis using ecological momentary assessment data. *European Psychiatry*, 66(Suppl 1):S750, 2023.

Appendix A

Detailed Implementation of Core Algorithms

A.1. Computing match and drop costs for DropDTW

```
def compute_all_costs(series1, series2, drop_cost_type="constant",
                      drop_multiplier=1.0, percentile=95,
                      cost_network=CostNetwork):
    """Compute alignment costs between time series"""
    # Validate inputs
    if not isinstance(series1, np.ndarray) or not isinstance(series2, np.
       ↪ ndarray):
        series1, series2 = np.array(series1), np.array(series2)

    # Compute pairwise cost
    zx_costs = np.zeros((len(series1), len(series2)))
    for i in range(len(series1)):
        for j in range(len(series2)):
            zx_costs[i, j] = np.sqrt(np.sum((series1[i] - series2[j])**2))

    # Compute drop costs based on strategy
    if drop_cost_type == "constant":
        x_drop_costs = np.ones(len(series1)) * drop_multiplier
        z_drop_costs = np.ones(len(series2)) * drop_multiplier

    elif drop_cost_type == "percentile":
        threshold = np.percentile(zx_costs, percentile)
        x_drop_costs = np.ones(len(series1)) * threshold * drop_multiplier
        z_drop_costs = np.ones(len(series2)) * threshold * drop_multiplier

    elif drop_cost_type == "learnable":
        mean1 = torch.tensor(np.mean(series1, axis=0), dtype=torch.float32)
        mean2 = torch.tensor(np.mean(series1, axis=0), dtype=torch.float32)

        with torch.no_grad():
            x_drop_costs = cost_network(mean2).item()*np.ones(len(series1))
            z_drop_costs = cost_network(mean1).item()*np.ones(len(series2))

    else:
        raise ValueError(f"Unknown drop_cost_type: {drop_cost_type}")

    return zx_costs, x_drop_costs, z_drop_costs
```

A.2. DropDTW for one-dimensional time series

```
def double_drop_dtw(costs, drop_costs1, drop_costs2, contiguous=True):
    """DTW algorithm allowing drops out of both sequences."""
    N, M = costs.shape
    INF = 1e9

    # Initialize tables: states are [matched, drop2, drop1, both_dropped]
    D = np.zeros([N + 1, M + 1, 4])
    D[1:, 0, :] = INF
    D[0, 1:, :] = INF
    D[0, 0, 1:] = INF

    # Allow initial drops
    D[0, 1:, 1], D[0, 1:, 3] = np.cumsum(drop_costs2), np.cumsum(
        ↪ drop_costs2)
    D[1:, 0, 2], D[1:, 0, 3] = np.cumsum(drop_costs1), np.cumsum(
        ↪ drop_costs1)

    # Path tracking
    P = np.zeros([N + 1, M + 1, 4, 3], dtype=int)
    for i in range(1, N + 1):
        P[i, 0, 2] = [i-1, 0, 2]
        P[i, 0, 3] = [i-1, 0, 3]
    for j in range(1, M + 1):
        P[0, j, 1] = [0, j-1, 1]
        P[0, j, 3] = [0, j-1, 3]

    # Fill tables
    for i in range(1, N + 1):
        for j in range(1, M + 1):
            # State 0: Both matched
            match_costs = [
                D[i-1, j-1, 0] + costs[i-1, j-1],  # matched
                D[i-1, j-1, 1] + costs[i-1, j-1],  # drop2
                D[i-1, j-1, 2] + costs[i-1, j-1],  # drop1
                D[i-1, j-1, 3] + costs[i-1, j-1]   # both dropped
            ]
            D[i, j, 0] = np.min(match_costs)
            P[i, j, 0] = [i-1, j-1, np.argmin(match_costs)]

            # State 1: Drop sequence 2
            drop2_costs = [
                D[i, j-1, 0] + drop_costs2[j-1],
                D[i, j-1, 1] + drop_costs2[j-1]
            ]
            D[i, j, 1] = np.min(drop2_costs)
            P[i, j, 1] = [i, j-1, np.argmin(drop2_costs)]

            # State 2: Drop sequence 1
            drop1_costs = [
                D[i-1, j, 0] + drop_costs1[i-1],
                D[i-1, j, 2] + drop_costs1[i-1]
            ]
            D[i, j, 2] = np.min(drop1_costs)
            P[i, j, 2] = [i-1, j, np.argmin(drop1_costs)]
```

```
            # State 3: Drop both
            both_costs = [
                D[i-1, j, 1] + drop_costs1[i-1],
                D[i, j-1, 2] + drop_costs2[j-1],
                D[i-1, j-1, 3] + drop_costs1[i-1] + drop_costs2[j-1]
            ]
            D[i, j, 3] = np.min(both_costs)
            P[i, j, 3] = [i-1, j, 1] if np.argmin(both_costs) == 0 else \
                         [i, j-1, 2] if np.argmin(both_costs) == 1 else \
                         [i-1, j-1, 3]

    # Backtracking
    cur_state = np.argmin(D[N, M, :])
    min_cost = D[N, M, cur_state]

    path = []
    i, j = N, M
    dropped1 = [N] if cur_state in [2, 3] else []
    dropped2 = [M] if cur_state in [1, 3] else []

    while not (i == 0 and j == 0):
        path.append((i, j))
        i_prev, j_prev, prev_state = P[i, j, cur_state]
        if prev_state in [1, 3] and j > 0:
            dropped2.append(j)
        if prev_state in [2, 3] and i > 0:
            dropped1.append(i)
        i, j, cur_state = i_prev, j_prev, prev_state

    path.reverse()

    return min_cost, path, dropped1, dropped2
```