Dynamic Time Warping Constraints for Semiconductor Processing

by

Rachel Owens

S.B., University of the Pacific (2020)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Rachel Owens
Department of Electrical Engineering and Computer Science
May 15, 2024

Certified by: Duane S. Boning
Clarence J. LeBel Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Dynamic Time Warping Constraints for Semiconductor Processing

by

Rachel Owens

Submitted to the Department of Electrical Engineering and Computer Science
on May 15, 2024
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering and Computer Science

Abstract

Semiconductor manufacturing processes have become increasingly complex with the continued growth of chip manufacturing. Monitoring these processes for anomalies is crucial for maintaining quality and yield. However, a notable challenge for monitoring time series signals are the nonlinear variations in signal timing. These small, but acceptable, temporal variations are typically caused by small run-to-run differences that are inherent to the process. Dynamic time warping (DTW) can be used for temporal alignment of signals, but is computationally expensive and prone to errors.

In this thesis, a new method is presented for preprocessing semiconductor fabrication sensor signals that improves anomaly detection model performance. The new method uses domain knowledge – specifically, process recipe step numbers – to create constraints that better align signals along the time dimension, that addresses this problem of nonlinear signal alignment. These constraints are tested on both synthetic as well as industrial datasets. The new step-constrained DTW is also extended as a distance measure for clustering time series.

Thesis Supervisor: Duane S. Boning
Title: Clarence J. LeBel Professor, Electrical Engineering and Computer Science

Acknowledgements

I find it hard to believe that two years have already passed since the day I received my acceptance letter to MIT. This has been one of the most fulfilling – and at times intimidating – endeavors I've ever undertaken, and I have frequently found myself humbled during the process. I would like to take this opportunity to express my deep appreciation for all the people who have assisted me, either directly or indirectly, on this journey.

First, I would like to express my gratitude to my advisor, Professor Duane Boning, for his invaluable guidance and encouragement throughout the entire research journey. His expertise, patience, and insightful feedback have been instrumental in shaping this work. I feel incredibly fortunate that I still have the rest of my Ph.D. years to study under such an exceptional mentor. I would also like to extend my heartfelt thanks to my friends and groupmates for their encouragement, stimulating discussions, and moral support, which have made this journey more joyful and meaningful.

Additionally, I would never have made it this far without the love and support of my family. Mom, Dad, and Elizabeth - thank you for always making me feel surrounded with love, even when I'm 3,000 miles away.

Finally, I must express my deepest appreciation to my boyfriend, Robert Ashby. His encouragement, support, and belief in me has been a source of motivation and solace during both the triumphs and tribulations of this endeavor. I am so incredibly grateful to have you in my life. This thesis is dedicated to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing complexity of semiconductor fabrication processes has enabled continued growth in chip manufacturing. Careful monitoring of these processes allows for improved quality and yield, which is crucial as the demand for semiconductors continues to increase. However, the data collected during monitoring has also increased in volume and complexity, and determining process conditions and quality has become more difficult. Unusual equipment sensor signal shapes can often indicate anomalous process conditions, which frequently lead to poor-quality products or even scrapped wafers. Detection of these anomalous wafer runs is therefore important to semiconductor manufacturers.

One notable challenge is that the timing of events in the signals can vary in a non-linear manner while still producing good-quality products. These nonlinear variations in signal timing make comparison of signals, and subsequently anomaly detection, difficult. Many recipes have an acceptable level of variation in the time it

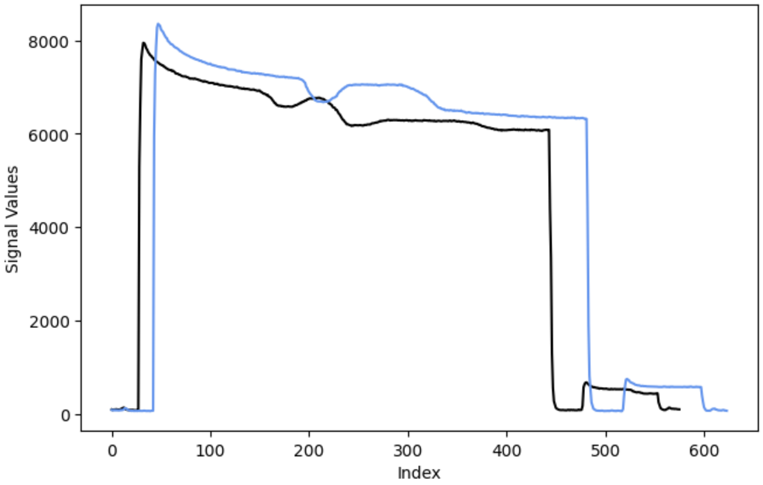


Figure 1.1: Example of the time variation in the endpoint signals of two nominal (acceptable) wafer runs.

takes to complete certain recipe steps. An example is displayed in Fig. 1.1. Process engineers are trained to discern such nominal or acceptable versus anomalous delays in processing, but it is not feasible for them to visually monitor all of the equipment data generated during processing. Therefore, automated anomaly detection methods need to be able to ignore acceptable amounts of time variation in equipment sensor data in order to detect genuine anomalies.

One approach is to align signals in the time domain prior to training, and inference with machine learning models. This is a flexible approach that can be used with many different datasets and models. This thesis explores using dynamic time warping for signal alignment preprocessing.

1.1 Motivation

Dynamic time warping (DTW) is a well-known algorithm for aligning signals along the time axis. While DTW can provide significant benefits, it can also be slow and prone to errors, particularly on anomalous signals. Many research efforts have focused on methods to minimize errors and speed up computation time [16, 27, 30, 36, 25]. However, none have addressed using additional semiconductor domain knowledge for improvements. This work uses domain knowledge of semiconductor processing to constrain the DTW algorithm in a manner that improves both speed and accuracy. By making use of the recipe step number signal provided by most processing equipment, we can introduce constraint boundaries and weighting that will encourage proper alignment of the processing signals.

Time series anomaly detection can use many different methods, and in this work the kernel density estimation (KDE) model [18] is used for testing the preprocessing methods. This method takes in good historical data and builds probability distributions of how those sensors normally behave. New samples can then be compared to these distributions to determine the likelihood that the sample came from the same nominal distribution. This process is visualized in Fig. 1.2.

However, even a small amount of time variation in the nominal signals used to train the model decreases the sensitivity of the model, as evidenced by how blurry the KDE distribution is in Fig. 1.2. This is why we want to improve signal alignment prior to training and inference.

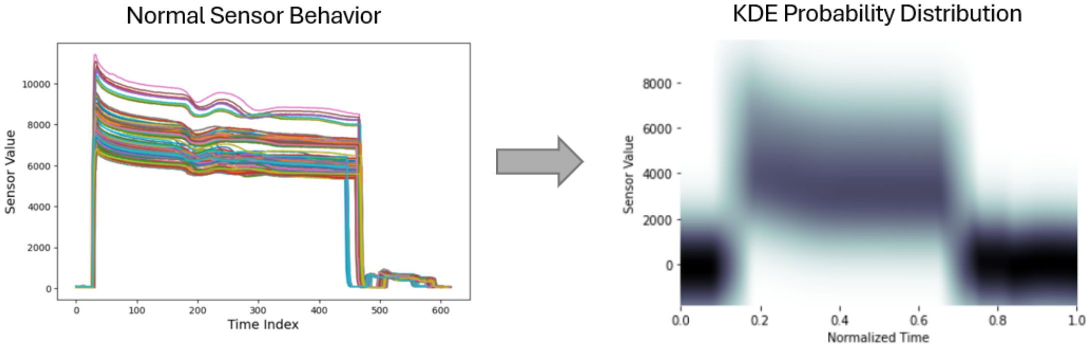


Figure 1.2: Construction of KDE empircal probability density estimates (right) based on representative known good sensor signals (left). The time variation contributes to wide (blurry) probability densities.

1.2 Thesis Structure

Chapter 2 introduces important background topics, including an overview of the DTW algorithm and several of its applications. The new constraints are developed in Chapter 3, followed by experimental setups and results in Chapter 4. Chapter 5 discusses additional uses for the new step-constrained DTW. Finally, Chapter 6 presents conclusions and possible topics for future work.

Chapter 2

Background and Related Work

In this chapter, necessary background knowledge is introduced. First is the definition of time series data, which is the format typically used for working with equipment sensor signals. Then the dynamic time warping (DTW) algorithm is reviewed, along with several of its variants and extensions.

2.1 Time Series Sensor Data

Time series are a data format that represents sequences of values, typically adhering to chronological ordering. Often generated by sampling real-world events, they are used across a wide range of different domains. A time series can be represented as $X \in \mathbb{R}^{N \times C}$, where C is the number of channels, each having length N. We will assume that each channel begins and ends at the same time and has an approximately constant sampling rate, ignoring any data loss that may occur in real-world applications.

When time series data are generated by similar events (e.g., a wafer processing event), comparison of two or more time series is often desired. However, what might be considered the “naive approach,” Euclidean distance, has many shortcomings when applied to time series. If one signal has even minor delays or distortions along the time axis, this can completely distort the similarity comparison. Therefore, the DTW algorithm, which performs a non-linear alignment of time series, is preferred for computing the distance between two time series.

2.2 The Dynamic Time Warping Algorithm

The DTW algorithm originated in the speech recognition field [32, 28] and was later introduced to the data mining community in 1994 by Berndt and Clifford [2]. DTW aligns two signals by stretching and compressing them in localized areas until they resemble each other as closely as possible. This alignment algorithm can also be used to calculate the dissimilarity or distance between the two signals. The DTW distance can be thought of as the “stretch-insensitive measure of the ‘inherent difference’ between two given time series” [10].

Suppose we have two time series that we want to compare: Q and C, of length n and m, respectively, where

$$Q = q_1, q_2, ..., q_i, ..., q_n, \tag{2.1}$$

and

$$C = c_1, c_2, ..., c_j, ..., c_m. \tag{2.2}$$

Note that the two time series do not need to be of equal length to perform DTW. We define a non-negative distance function f for any pair of elements q_i and c_j:

$$d(i, j) = f(q_i, c_j) \geq 0. \tag{2.3}$$

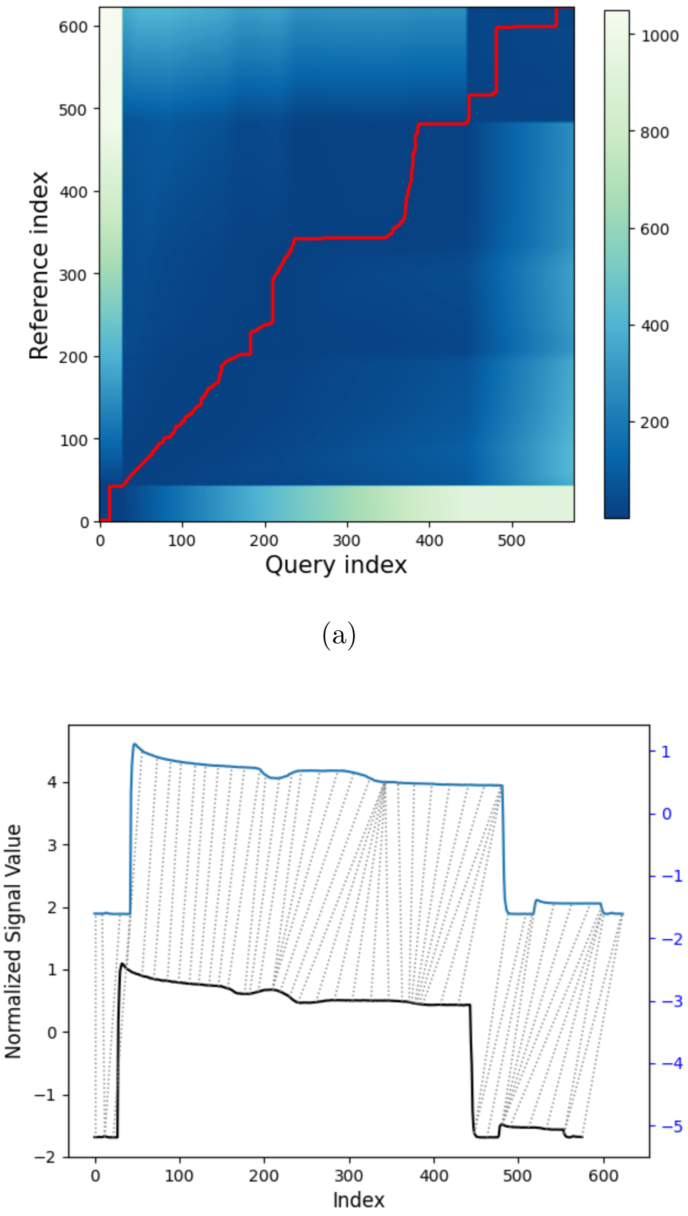
This distance function is used to calculate the distance between every point in Q with every point in C. Typically the Euclidean distance is used for the distance function, where $d(i, j) = (q_i - c_j)^2$. The calculated distances, or "costs," are stored in a matrix of size $n \times m$, often referred to as the cost matrix. A heatmap visualization of a cost matrix can be seen in Figure 2.1a. The only input to the DTW algorithm is this cost matrix, where each matrix element (i, j) is the defined distance between the ith point of Q and the jth point of C.

The warping path W traces the lowest-cost path through the cost matrix, starting from the $(1, 1)$ index and ending at (m, n). The path consists of a contiguous set of matrix elements that defines a mapping between Q and C. A single mapping element connects one point in Q to one point in C. Path W has length K, which must be at least as long as the longest time series but shorter than the sum of the two.

$$W = w_1, w_2, ..., w_k, ..., w_K \qquad \max(n, m) \leq K < m + n - 1. \tag{2.4}$$

The warping path is typically subject to several conditions.

- Boundaries: $w_1 = (1, 1)$ and $w_k = (m, n)$. This requires that the warping path must start at the beginning of both time series (bottom left of the cost matrix) and finish at the end of both time series (top right of the cost matrix).



(a)

(b)

Figure 2.1: Example of the original DTW algorithm, applied to two plasma etch endpoint signals. (a) The warp path (red line) is traced across the cost matrix. (b) Alignment mappings between the signals. Graphs were generated using the dtw-python package [10].

- Continuity: Given that $w_k = (a, b)$, then $w_{k-1} = (a', b')$ where $a - a' \leq 1$ and $b - b' \leq 1$. This means that allowable steps in the warping path are restricted to adjacent cells (including diagonally adjacent), and every index of each time series must be used.

- Monotonicity: Given that $w_k = (a, b)$, then $w_{k-1} = (a', b')$ where $a - a' \geq 0$ and $b - b' \geq 0$. This means that points in the warping path must be monotonically spaced in time, which ensures that the warping path does not overlap itself.

While there are many warping paths that satisfy these constraints, the optimal warp path is the one that minimizes the warping cost [2]:

$$DTW(Q, C) = \min \sum_{k=1}^{K} d(w_k). \tag{2.5}$$

This path can be found efficiently using dynamic programming in $O(nm)$ time. A visualization of the warp path and mappings can be seen in Figure 2.1. We see that the warp path (plotted in dark blue) follows the low-cost 'valley' through the cost matrix. Next to it in Figure 2.1b are the original signals, with the mappings between them plotted in light gray.

The original implementation of DTW does not restrict the boundary of the warping path within the cost matrix – this is known as unconstrained DTW. A variation is constrained DTW, which limits the possible paths that can be taken through the cost matrix.

2.3 DTW Constraints

Constrained DTW helps to avoid pathological mappings between two time series, also known as singularities. This occurs when a single point in one of the time series is mapped to far too many points in the other time series, resulting in poor alignment.

Many constraint methods have been proposed to reduce these singularities; these methods can be loosely grouped into step pattern and windowing approaches. Step patterns, also called local slope constraints, limit the direction of transitions between matrix elements. This results in limiting the amount of time stretch and compression that is allowed at any local area of the alignment. Common step pattern constraints include those introduced by Myers in 1980 [23] and Rabiner and Juang in 1993 [26].

Window constraints define a window within the cost matrix where the warping path is free to take any shape. Popular versions of window constraints include Sakoe-Chiba bands [29] and Itakura parallelograms [12]. Sakoe-Chiba bands generally perform better than Itakura parallelograms, but there are still individual datasets where Itakura can produce superior results [9]. This indicates that the proper constraint method is highly dependent on the dataset. Additionally, the optimal size of the constraint window is dependent on dataset properties such as dataset size and time series shapes [6]. Constraints can also be adapted to reflect structural characteristics of the dataset or time series [27, 3].

2.4 Time Series Averaging

While the concept of “average” is easily defined for many data types, for time series the task of finding the average of a group of series is more complex. A typical inclination may be to use the Euclidean average at each time point, but this can obscure temporal features that are characteristic of the series. A major improvement for the averaging of time series came from Petitjean et al. and is known as DTW barycenter averaging (DBA) [24]. DBA iteratively refines an average series using a two-step expectation-maximization scheme. The process first computes the DTW distance between the temporary average sequence and all the other sequences in order to find the mappings between their coordinates. Then it updates each coordinate of the average sequence to be the barycenter average of all the coordinates that were associated to it in the first step. DBA preserves the shape of the time series, and can be used as a representative of a group of time series. For this reason it is often used as a template or references series, or for clustering time series.

Another method for time series averaging uses Fréchet means, which rely on differentiation to find global minimums. One shortcoming of DTW is that it is not differentiable with respect to its inputs due to the min operator being non-differentiable. This is addressed by the soft-DTW formulation of DTW, which achieves differentiability by replacing the min with a soft-min operator [4]. Soft-DTW is defined as

$$\mathrm{dtw}_\gamma(x, y) := \min{}^\gamma\{\langle A, \Delta(x, y)\rangle, A \in \mathcal{A}_{n,m}\}, \tag{2.6}$$

where the generalized min operator is

$$\min{}^{\gamma}\{a_1, ..., a_n\} := \begin{cases} \min_{i \leq n} a_i, & \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0. \end{cases} \tag{2.7}$$

This formulation shows that soft-DTW depends on a hyperparameter γ that controls how much smoothing occurs in the resulting metric. The original DTW distance can be calculated by setting γ to 0.

Since soft-DTW is differentiable, a barycenter average of time series can be calculated by directly applying Fréchet means with respect to the soft-DTW algorithm. The use of the soft-min operator smoothes out local minima, which provides a better optimization landscape compared to DBA, which can sometimes suffer from getting stuck in local minima. The average time series found using soft-DTW are typically smoother than using DBA.

2.5 DTW for Clustering

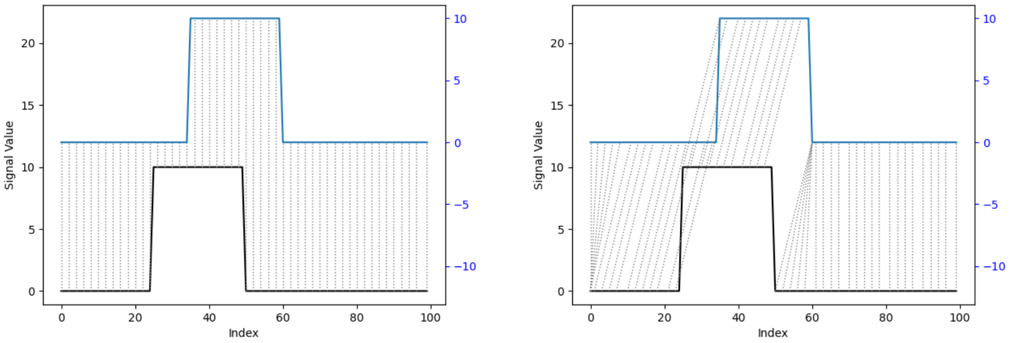
Clustering is a method of unsupervised pattern recognition, with the goal of grouping similar data samples into clusters. This is done by defining a distance measure between pairs of points in the data sample, and then minimizing the within-cluster distance and maximizing the between-cluster distance. The more distinct the clusters are, the better the clustering quality is considered to be. DTW can be used as the distance measure for clustering time series, because the output of the algorithm gives a measure of the dissimilarity between two sequences. While the DTW-distance is not a formal metric distance due to its inability to satisfy the triangle inequality and the identity of indiscernibles [31, 13], it can still be used effectively as a distance measure.

Much work has been done on using dynamic time warping and clustering for time series classification, and the general consensus that has emerged is that the k-nearest-neighbor algorithm (kNN) is the most accurate classifier, even across different domains [25, 35]. However, kNN is a supervised method which requires labelled data, so for applications that require unsupervised learning other algorithms such as k-means are also frequently used [15, 21].

An insightful paper by Petitjean et al. [25] highlights the usefulness of DBA for clustering. They demonstrate that using DBA to condense the training dataset can bring improvements in both accuracy and speed, which is especially beneficial for resource-constrained applications. DBA has also been used in the domain of plasma etching for building reference signals for clustering [11] and for augmenting sensor readings with soft labelling [19].

2.6 DTW for Anomaly Detection

Anomaly detection is an important task in semiconductor manufacturing. With the increasing complexity of processes and devices, being able to monitor and detect faults quickly is critical to maintain profitability. Anomaly detection problems in time series are frequently formulated to compare differences between a new, unlabelled sample and established normal behavior. Detection methods typically rely



(a) Euclidean alignment

(b) DTW alignment

Figure 2.2: The DTW distance measure is able to handle distortions in the time axis, unlike the Euclidean distance measure.

on measuring distances between time series. However, standard distance measures such as the Euclidean distance fall apart quickly when applied to time series. Even minor distortions in the time axis can cause the distance measure to dramatically increase. A simple example using a shifted step signal is visualized in Fig. 2.2. The DTW alignment is able to map the shifted signal correctly and has a distance of zero. The Euclidean distance, on the other hand, is only able to use a one-to-one mapping which cannot handle the time shift, and thus has an increased distance of 45 for this particular example. Therefore, elastic similarity measures, such as DTW, are much preferred when working with time series [20].

Three main types of anomalies occur in time series data:

- Point: individual data points that are unusual when compared to other values in the time series. These can be global or local outliers.

- Subsequence: a section of the time series that is unusual within the context of the larger pattern.

- Collective: an entire time series that is unusual when compared to normal behavior in other time series.

A detection method is usually most sensitive to one type of anomaly, although some can be used to detect all types.

DTW has been applied to time series anomaly detection in several ways. A frequent method is to use DTW as the distance function for other algorithms, as in [22, 14, 34]. Another method is to align the time series prior to calculating individual point distances [7, 33]. Anomalous signals can also be detected from the information obtained while creating the warping path [17].

Chapter 3

Methods

In this chapter new DTW constraints are developed that incorporate knowledge of the processing recipe step transitions. The first section discusses shortcomings of existing DTW methods. The next three sections detail how DTW constraints can be designed to incorporate the recipe steps, from the generation of reference signals, to the specific constraints design, and finally the application to signal alignment.

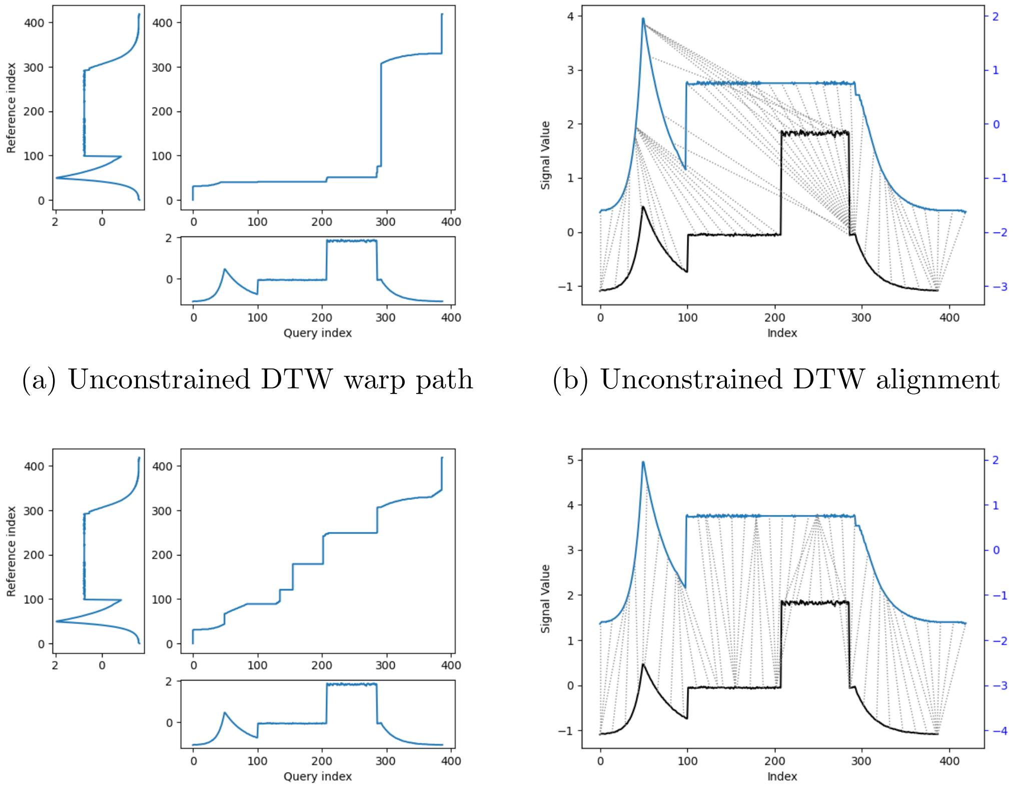
3.1 Shortcomings of DTW

While the original DTW algorithm is popular and effective in many scenarios, it still has shortcomings. As discussed in Section 2.3, one of the more common problems is known as a singularity, where an unintuitive one-to-many mapping occurs, resulting in poor alignment. Different constraint methods have been proposed to limit the occurrence of singularities, such as the popular Sakoe-Chiba method.

The warp paths for both the original unconstrained DTW and the Sakoe-Chiba constrained method are shown in Figure 3.1. Extreme singularities can be seen in the unconstrained DTW warp path, as evidenced by the straight horizontal and vertical lines indicating unintuitive mappings. The singularities are smaller but still present in the algorithm constrained with Sakoe-Chiba bands. These singularities can distort the resulting warped signals, causing poor alignment with the reference signal. The alignment is also inconsistent across many samples, particularly when anomalies are present. This confirms the need for an improved alignment method.

3.2 Generation of Reference Signals

The DTW algorithm optimally aligns two signals at a time; however, our goal is to align a group of signals. Therefore, we establish a reference signal for each sensor that other signals can be compared to. If “golden” reference signals are available they are ideal to use, but that is not common. To solve this, we generate reference signals from known-good data that was previously collected from the same tool and recipe. DTW barycenter averaging (DBA) is used, which iteratively creates an “average” time series using a two-step expectation-maximization scheme [24]. DBA preserves the shape of the time series and can be used as a representative of a group of time series, which makes it useful for generating template or reference series. It has been shown to perform better on plasma etch signals than comparison methods such as soft-DTW [11].



(a) Unconstrained DTW warp path

(b) Unconstrained DTW alignment

(c) Sakoe-Chiba constrained DTW warp path

(d) Sakoe-Chiba constrained DTW alignment

Figure 3.1: DTW warp paths and signal alignments for unconstrained DTW and Sakoe-Chiba constrained DTW. Both methods still suffer from singularity problems, indicated by the abrupt path changes and unintuitive mappings.

3.3 Constraints

We propose using semiconductor processing recipe step numbers to develop constraints for the DTW algorithm. Characteristic features in the signals typically occur when a recipe transitions from one step to the next, so we want to encourage alignment between signals at recipe step transition points. Signal features, however, do not necessarily align precisely with step increments. For example, the endpoint signal is commonly used to detect the endpoint of certain reactions, which then indicates it is time to proceed to the next recipe step. A signal feature might therefore occur several time points *before* the step transition point. Certain other recipe steps call for tool adjustments that induce changes in the process environment, resulting in distinct features in the measured signals. These kinds of adjustments would cause signal features to appear *after* the step transition point.

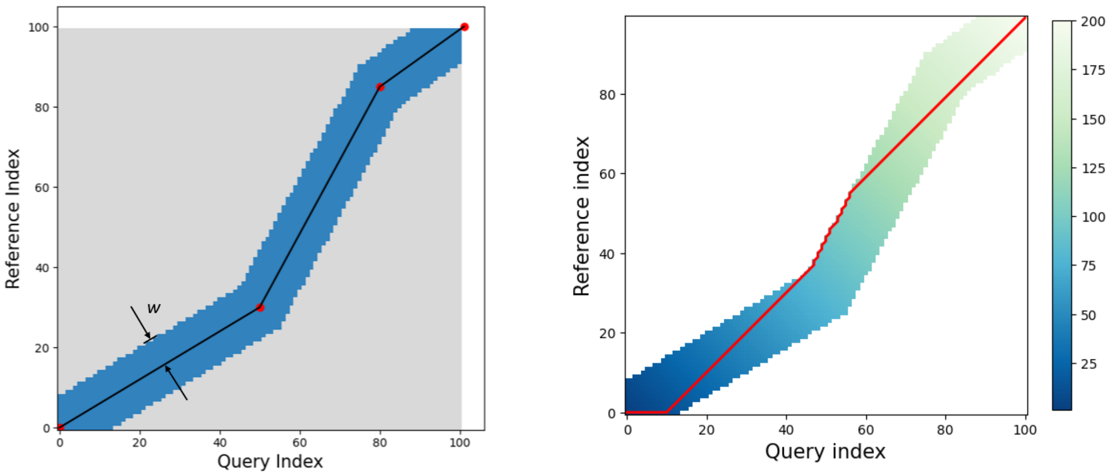
For these reasons, the time point at which the recipe step transition occurs only gives an approximate time location for possible signal features. While additional subject matter expertise could be used to identify specific features for alignment, this would require significant human effort for each process and recipe, which is infeasible in most fabs. Therefore, using the approximate time location for features is preferred, as this method can be applied in an unsupervised manner.

The step transition points are used to develop adaptive constraints that reflect the high-level recipe structure inherent in these types of signals. To be able to align the incoming sensor signals in an unsupervised fashion, a reference signal is generated for each sensor from known-good data using the DBA method discussed in Section 3.2. The reference signal will be used as a guide, and the new data will be warped to align with the reference.

To create the constraints, the step transition points are first extracted from the step signal. These time points are plotted on the cost matrix, and a virtual line is drawn between the points. An envelope is then built around this virtual line which will be the allowed region of the constrained matrix. The boundary of the envelope is a distance w from the virtual line. This constraint width is set as a hyperparameter, and controls how much time stretch is allowed by the constraints. A visual example of building the constraints can be seen in Figure 3.2a. Once the constrained area is defined, it is applied to the cost matrix to set boundaries on the warp path, as shown in Figure 3.2b. By restricting the allowed paths through the cost matrix, we can reduce the singularities that cause poor alignment.

3.4 Signal Alignment for Preprocessing

DTW can be a useful preprocessing step for time series data prior to use in machine learning models. There is increasing interest in applying this concept to monitoring semiconductor processes [1, 5, 8], since aligning signals that might have variation in the time axis can improve performance of the models. Using DTW to find the optimal alignment between a reference signal and query signal allows one to apply those alignment mappings to create the warped query signal. Given the warp path W that describes the alignment mappings, the warped signal for either the reference or query signal can be found by selecting the corresponding sensor value for each element in W. To create the warped query signal, one iterates through the alignment



(a) Constraints are built using the recipe step transition points (red points). A virtual line (black) connects the points, and then an envelope (blue) of width w is built around the line, which will be the allowed area of the constraint. The grey area is out-of-bounds.

(b) Constraints derived from the recipe steps are imposed on the DTW cost matrix. The warp path (red line) traces the path of optimal alignment between the two signals within the constrained area.

Figure 3.2: Building and using constraints on the DTW cost matrix.

elements to locate the query index in the mapping $w(a, b)$. The query index then provides the original query signal value, which is appended to the warped query.

Aligning all the samples in a dataset to the same reference signal reduces the time-variability of characteristic features. DTW was specifically mentioned as a possible preprocessing extension for the kernel density estimation (KDE) method [18], and this is the primary anomaly detection model for which the DTW preprocessing approach will be tested in this work.

Chapter 4

Experiments and Datasets

This chapter demonstrates the effectiveness of the new step-constrained DTW method for signal alignment preprocessing. The proposed method is compared to five other alignment methods on two different datasets, with additional exploration into the differences between methods.

4.1 Datasets

4.1.1 Synthetic Dataset

To demonstrate the proposed method, a synthetic dataset is generated that mimics the sensor signals that occur on semiconductor processing equipment. A visualization of this dataset is shown in Figure 4.1. Each “run” contains four sensor signals as well as a step signal, all with three recipe steps. The length of each step is normally distributed and equal across all sensors, causing changes in the signals exactly at the transitions of steps. Thus, the total length of the sensors may vary from one run to the next, but the overall shape remains the same and would be considered a nominal or acceptable process run. Anomalies are injected by randomly scaling sections of runs.

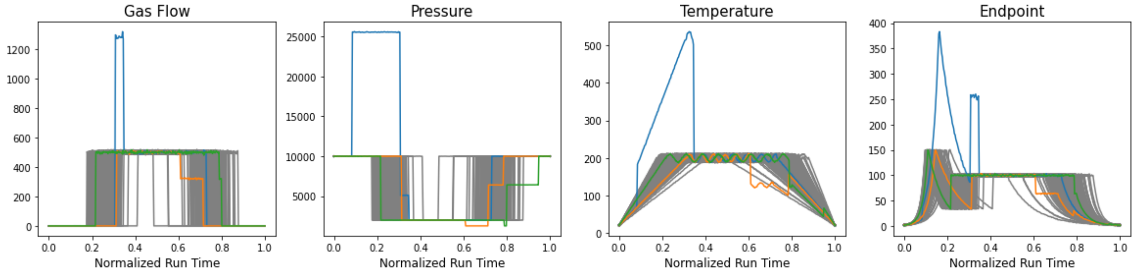


Figure 4.1: Synthetic dataset with four sensor signals that mimic a generic plasma etch. An example training set of nominal data is shown in grey, with three anomalous runs displayed in color. Some runs (e.g., green signals) are subtle anomalies in just one of the four sensor signals for that run; others (e.g., blue signals) are relatively large or “easy to detect” anomalies spanning all four signals for that run.

4.1.2 Industrial Dataset

A historical plasma etch dataset from Analog Devices, Inc. is used that includes both nominal data and anomalous data from a process fault. There are 2004 nominal runs and 342 anomalous runs, for a total of 2,346 labelled wafer runs. The dataset contains 31 different sensor signals, of which five have been selected for testing due to their relevance: RF power, gas flow, chuck temperature, OES endpoint, and chamber pressure. A normalized sample of this dataset is displayed in Figure 4.2, along with the recipe step signals.

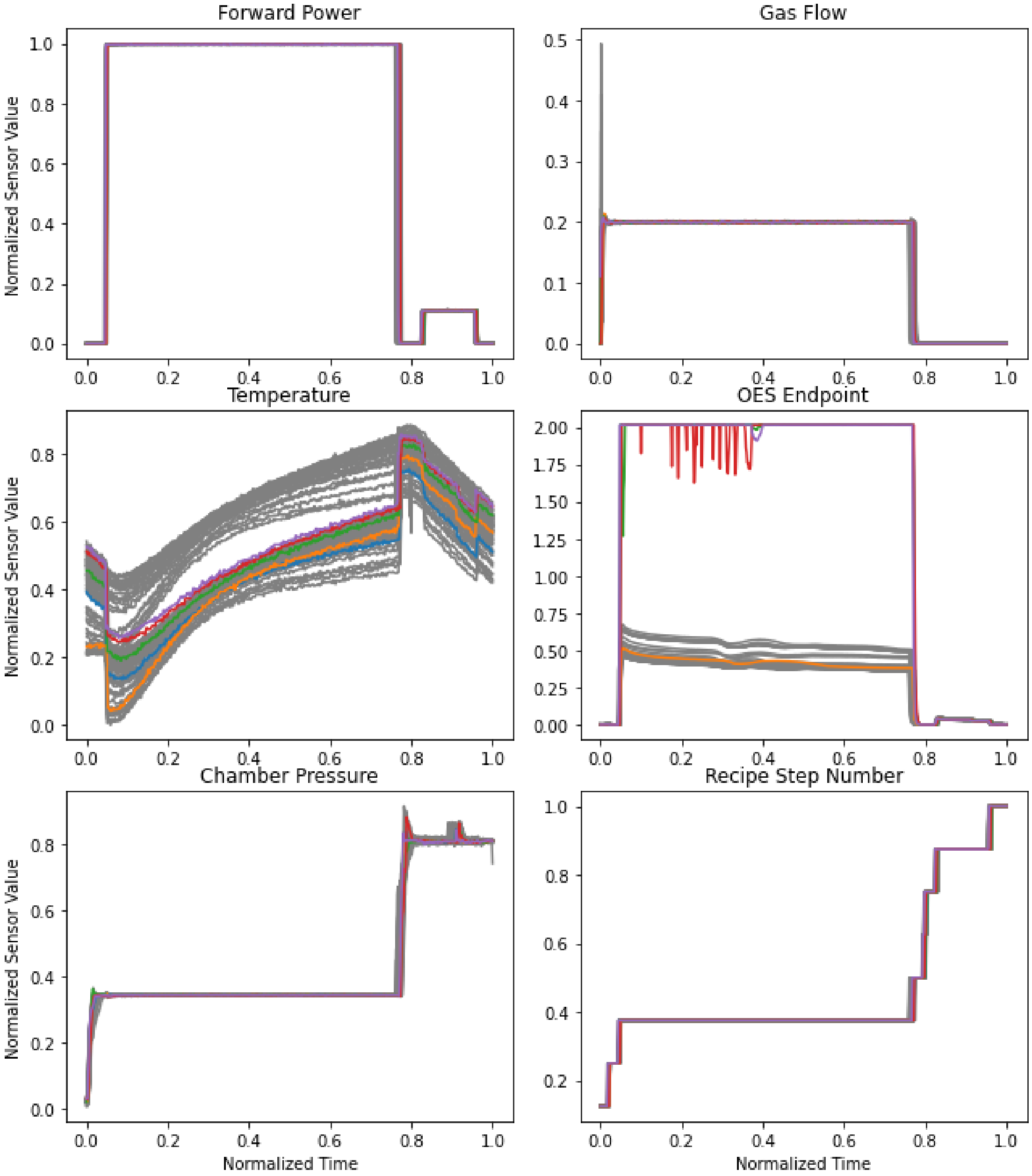


Figure 4.2: Plasma etch dataset with six recorded sensor signals. An example training set of nominal data is shown in grey, with five anomalous runs displayed in color.

Dataset Sub-sampling

The proposed methods are tested on two labelled datasets: one synthetically generated and one real plasma etch dataset. In order to understand how the methods generalize between similar datasets drawn from the same underlying distribution, we have sub-sampled from these two primary datasets to create smaller training sets from each primary dataset. To create the testing subset, 100 nominal runs and 100 anomalous runs are set aside and used for every evaluation. From the remaining nominal data, runs are sampled to create the training subset. This sampling is repeated to create different training sets, and the methods are tested on all and averaged in order to understand the robustness of the methods.

4.2 DBA vs. Soft-DTW

To build the reference signals needed for DTW, two popular time series averaging methods are selected for comparison. These methods are applied to the training datasets, which contain only known-good data. The reference time series obtained will thus be representative of known-good runs. Since there are typically multiple sensors, multivariate DTW will need to be used, with a reference signal created for each sensor.

Both methods – DBA and soft-DTW barycenters – are tested on a plasma etch dataset. An experiment is performed to determine which method creates a better average of a group of time series. First, 19 sensors are selected from the known-good plasma etch dataset. The data is z-score normalized to account for the different

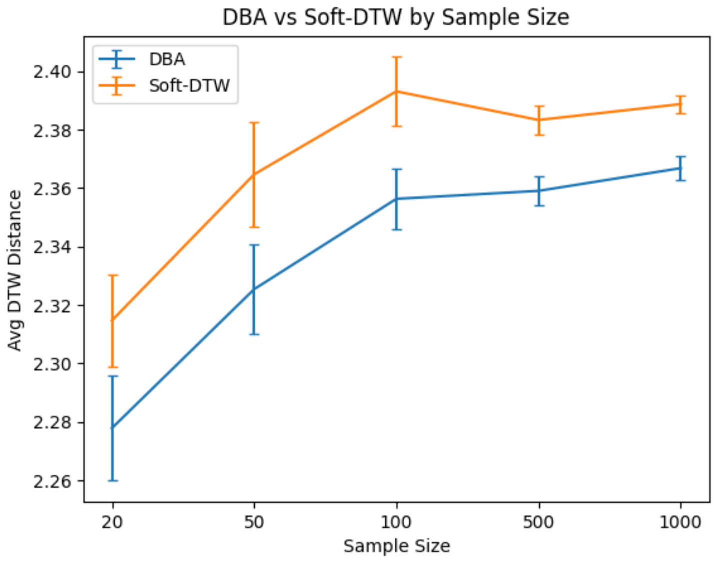


Figure 4.3: The best time series averaging method for minimizing DTW distance is DBA. Error bars represent one standard error of the mean.

sensor value scales. Next, a randomized dataset (of a determined sample size) is selected for each sensor, and both averaging methods are used to find the average time series for that sensor, creating a reference sequence for each method. The gamma hyperparameter for soft-DTW is set to $\gamma = 0.1$, based on a similar prior experiment that suggests it has little impact on the mean DTW distance [11]. Then the DTW distance is measured between each sample in the dataset and the newly created reference sequence. For each determined sample size, this process is repeated 15 times and the DTW distance results are averaged. A plot of the overall results can be seen in Figure 4.3. For all sample sizes, DBA achieves a lower average DTW distance, thus outperforming the soft-DTW method. This agrees with the previous work by He [11].

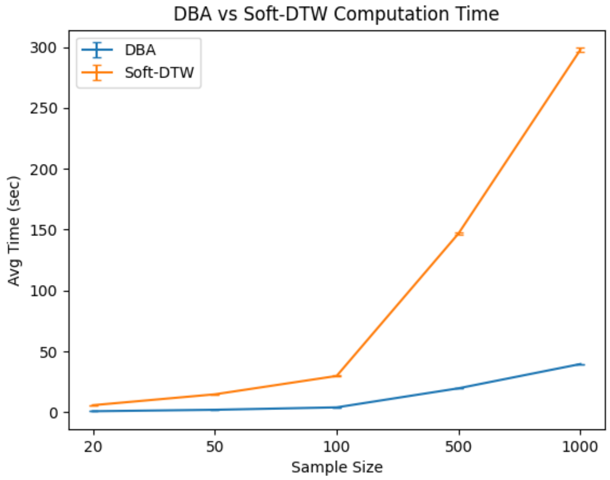


Figure 4.4: DBA requires significantly less computation time than soft-DTW. Error bars represent one standard error of the mean.

Another consideration for choosing a barycenter averaging method is the computation time. As shown in Figure 4.4, the computation time for soft-DTW grows exponentially with the size of the dataset. Therefore, it would be preferable to use DBA if computation resources are limited or the target dataset is large.

Additionally, since the reference signal will be used to align new data (that was not used to create the average), we are interested in how well it generalizes as a representative of unseen data. For this experiment, both DBA and soft-DTW create reference signals by averaging a small set of samples. Then the DTW distance is measured between these generated reference signals and 1000 randomized, known-good signals. This is repeated 15 times for each chosen sample size and averaged, and the results can be seen in Figure 4.5.

Soft-DTW generalizes better than DBA when starting with a small dataset. However, for datasets larger than approximately 20 samples, DBA performs better. The

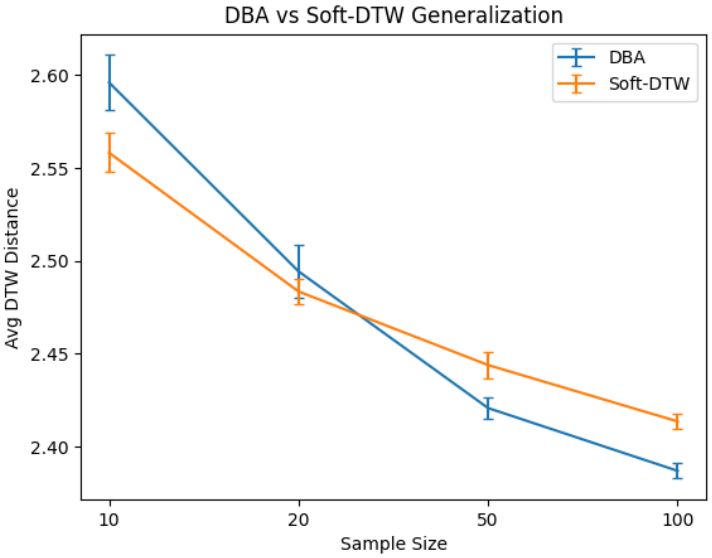


Figure 4.5: Soft-DTW does a better job of representing small datasets, while DBA performs better with larger datasets. Error bars represent one standard error of the mean.

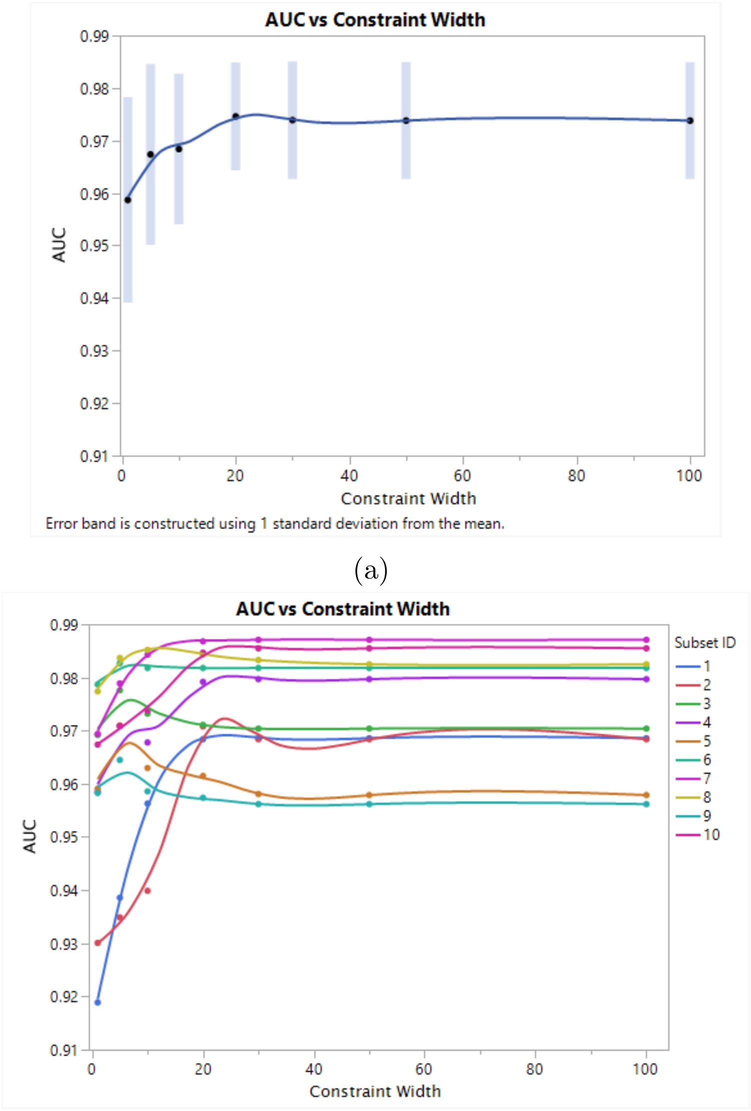
results from these larger sample sizes indicate that DBA might be overfitting to a particular group of samples when smaller sample sizes are involved. Soft-DTW smoothes the optimization landscape, whereas DBA does not perform this smoothing and so adopts some of the idiosyncrasies of those particular samples. This effect would be heightened for smaller sample sizes. Therefore, if working with small datasets of fewer than approximately 20 samples, soft-DTW would be an appropriate method to use.

4.3 Tuning Width Parameter

The width parameter w reflects how much stretch is allowed in the time alignment of the warped signals. Too small, and it restricts the allowed area, possibly blocking the optimal alignment path; too large, and the allowed region is overly wide, which can lead to excessive singularities and worse alignment. Therefore, it is important to establish an acceptable range for w. The optimal value can vary from dataset to dataset, which was confirmed by Dau et al. [6]. The experiment below serves as a guide for finding a useful range for a particular dataset, as well as an exploration of how non-ideal width values impact alignment.

Smaller datasets are built using the sampling method described in Section 4.1.2 to create ten subsets from the plasma etch dataset. The width value w is varied between the following values: 1, 5, 10, 20, 30, 50, and 100. A value of one restricts the allowed path to a straight line between step transition points, while larger values allow for a wider allowed area, and thus more options for warp paths.

We use the area Under the ROC (receiver operating characteristic) curve (AUC-



(a)

(b)

Figure 4.6: (a) Average anomaly detection AUCs using the step constrained DTW preprocessing method while varying the width hyperparameter. (b) AUC scores for individual data subsets using the step constrained DTW preprocessing method while varying the width hyperparameter. There is large variation across datasets in the response to varying the constraint width.

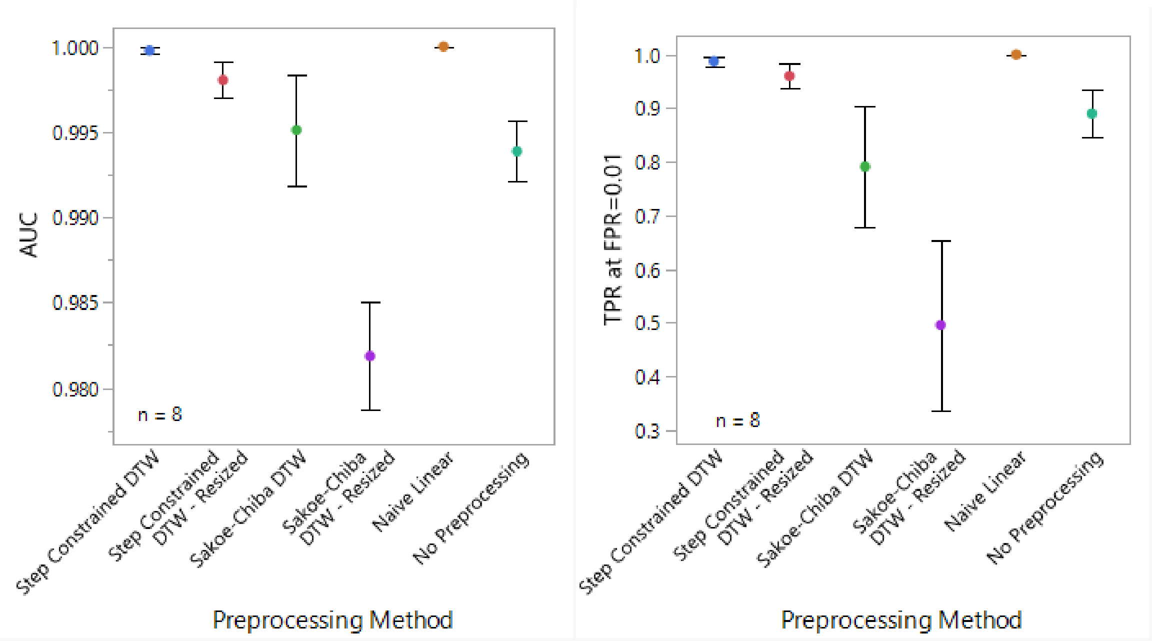
ROC) metric, which measures the performance of classifier models. A perfect score of 1 indicates that the model is able to correctly classify every sample. The AUC score is observed for the KDE method across values of w, and the results are shown in Figure 4.6. The highest performance is achieved using $w = 20$, which has the highest average AUC value across all data subsets, and corresponds to about 3-4% of the total length of a time series in the plasma etch dataset. However, it is important to note that there is large variation between datasets in the response to varying the constraint width. In particular, several subsets show a significant drop in performance when the width is decreased below 20. Therefore, if tuning is not completed for a future application, it would be better to err on the higher side rather than risk the decreased performance seen with a tight constraint on some subsets.

4.4 DTW Constraints

We compare the performance of six different methods for use in signal alignment preprocessing, when coupled to KDE anomaly detection. The first method, which we call "no preprocessing," does not alter the signal at all, and assumes that the data points are equally spaced in time, with the first point starting at time zero and the last point occurring at time equal to one. The new constraints, as described above, are another method. However, the warped signals end up being longer than the original signal, which increases computation time during model training and evaluation. To account for this, we also test resizing the warped signals down to the original size using linear interpolation. Another DTW constraint method, the Sakoe-Chiba band, is also tested in both the full and resized versions. Finally, a naive linear time scaling method that makes use of the recipe step transition points is also compared. It linearly rescales each step in time to match the length of the reference signal step.

The main hyperparameter for the KDE method [18], Δt, is tested across three different values: 0.005, 0.01, and 0.02. The preprocessing methods may perform optimally under different hyperparameter conditions, so for a fair comparison we choose the best-performing conditions for each method and average those results. We again use the AUC-ROC score, as well as the true positive rate (TPR) of the classifier, given an acceptable false positive rate (FPR) of 1%. This is an important metric for manufacturing because it means that, statistically, 1% of the nominal runs will cause a false alarm that engineers will have to review. Decreasing this further would require a tradeoff with the sensitivity of the model, possibly decreasing the number of true anomalies that the model is able to detect.

The results for the synthetic dataset are shown in Figure 4.7. The best method in this experiment is the naive linear rescale, with our step constraints a close second and much better than the rest. This result makes sense for this dataset, because we know the exact time that the recipe step changes and that it occurs at the exact same time that the signal features change. Therefore, the naive linear rescaling is able to perfectly align the signals. This demonstrates that if we can achieve perfect alignment, the KDE method is able to correctly discriminate between the nominal and anomalous signals for this synthetic dataset. However, in real-world data, the step transition points and signal features do not have this perfect alignment. This is why our step constraint method is better than linear rescale, as shown in the next



(a)

(b)

Figure 4.7: (a) Average anomaly detection AUCs for the different preprocessing methods on the synthetic dataset. (b) Average true positive rates at a false positive rate of 1% for anomaly detection with the different preprocessing methods on the synthetic dataset. Error bars show one standard deviation.

result.

The plasma etch dataset provides an opportunity to apply these methods to a real-world scenario. Two optical endpoint signals from this dataset were displayed previously in Figure 2.1b, which shows the more subtle variation in recipe step lengths. The results for AUC and the true positive rates at a false positive rate of 1% are given in Figure 4.8. Note that the naive linear rescaling performs much worse on this dataset. This is because real-world data is inherently messy, and we do not know the exact moment that step transitions and signal features occur. By forcing the signals to align at points that may be non-ideal, it actually performs worse than the more

	AUC		TPR at FPR=0.01	
Preprocessing Method	Mean	Std Dev	Mean	Std Dev
Step Constraints	0.9998	0.0002	0.9875	0.0089
Step Constraints - Resized	0.9980	0.0010	0.9600	0.0233
Sakoe-Chiba	0.9951	0.0032	0.7913	0.1131
Sakoe-Chiba - Resized	0.9819	0.0031	0.4963	0.1583
Naive Linear	**1.000**	0.00	**1.000**	0.00
No Preprocessing	0.9939	0.0018	0.8900	0.0441

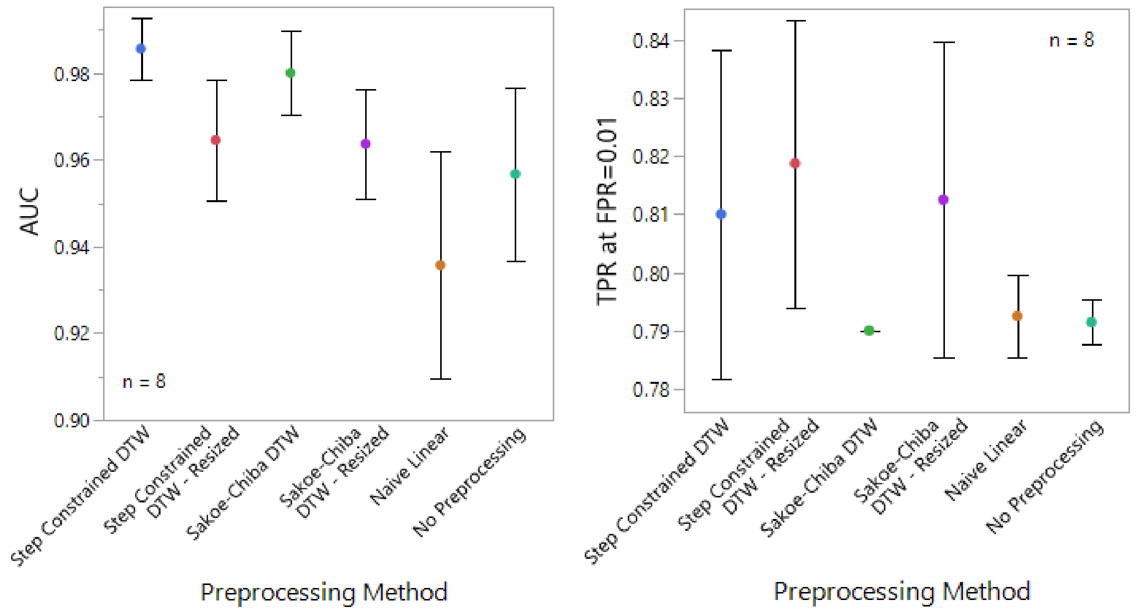
Table 4.1: Average anomaly detection AUCs and TPRs for the different preprocessing methods on the synthetic dataset. Best performance is bolded and second-best is underlined.

	AUC		TPR at FPR=0.01	
Preprocessing Method	Mean	Std Dev	Mean	Std Dev
Step Constraints	**0.9856**	0.0072	0.8100	0.0283
Step Constraints - Resized	0.9645	0.0139	**0.8188**	0.0247
Sakoe-Chiba	0.9800	0.0096	0.7900	0.00
Sakoe-Chiba - Resized	0.9637	0.0127	0.8125	0.0271
Naive Linear	0.9357	0.0262	0.7925	0.0071
No Preprocessing	0.9567	0.0199	0.7914	0.0038

Table 4.2: Average anomaly detection AUCs and TPRs for the different preprocessing methods on the plasma etch dataset. Best performance is bolded and second-best is underlined.

flexible approaches.

The best-performing method is the new step constraints, with the Sakoe-Chiba band a close second. The Sakoe-Chiba method appears to perform worse when the time length variation in recipe steps is more extreme, as was the case with the synthetic dataset. Since the Sakoe-Chiba band does not take into account the recipe steps, it cannot align those sensor signals quite as well. Figure 4.9 explores how the differences in warp path affect the Sakoe-Chiba band constraints, as opposed to



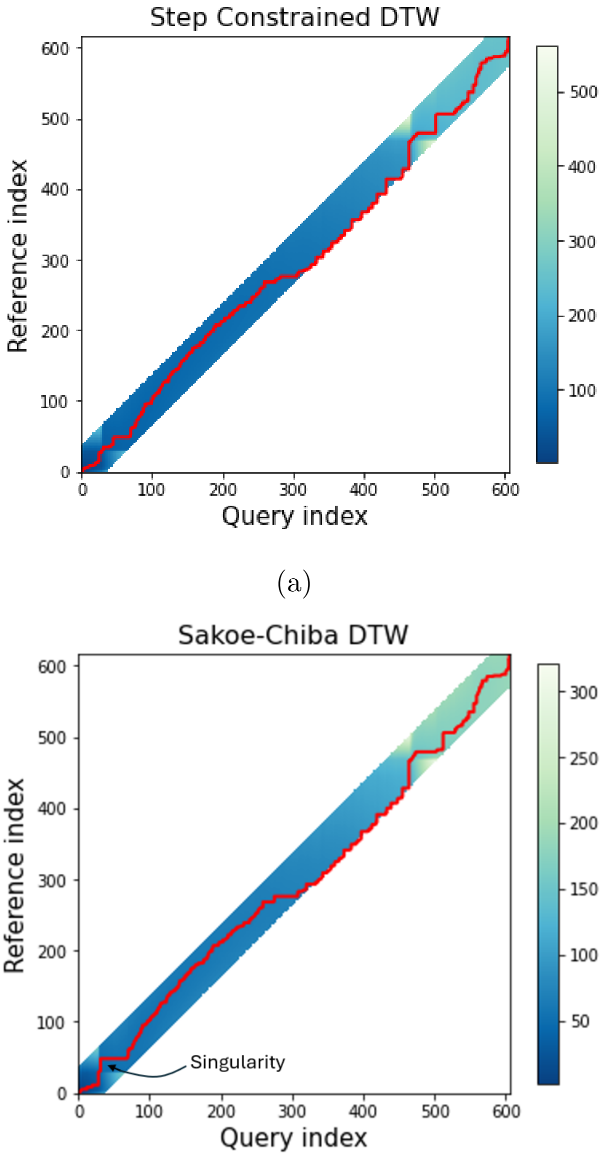
(a)

(b)

Figure 4.8: (a) Average anomaly detection AUCs for the different preprocessing methods on the plasma etch dataset. (b) Average true positive rates at a false positive rate of 1% for anomaly detection with the different preprocessing methods on the plasma etch dataset. Error bars show one standard deviation.

the step constrained DTW. The Sakoe-Chiba preprocessing misses a small number of subtle anomalies, which appears to be due in part to singularities in the warp path. However, it would still be a viable method to use if the step signal was unavailable for some reason. The resized methods do not perform as well as the full-length warped signals, but they do still offer a small improvement over no preprocessing. Additionally, both resized methods provide a higher TPR than the full-length signals do, which could be of practical use for industry implementation. There may be a length in between these sizes which maximizes the benefits of the different lengths with minimal increase in computation time, which could be explored in future work.

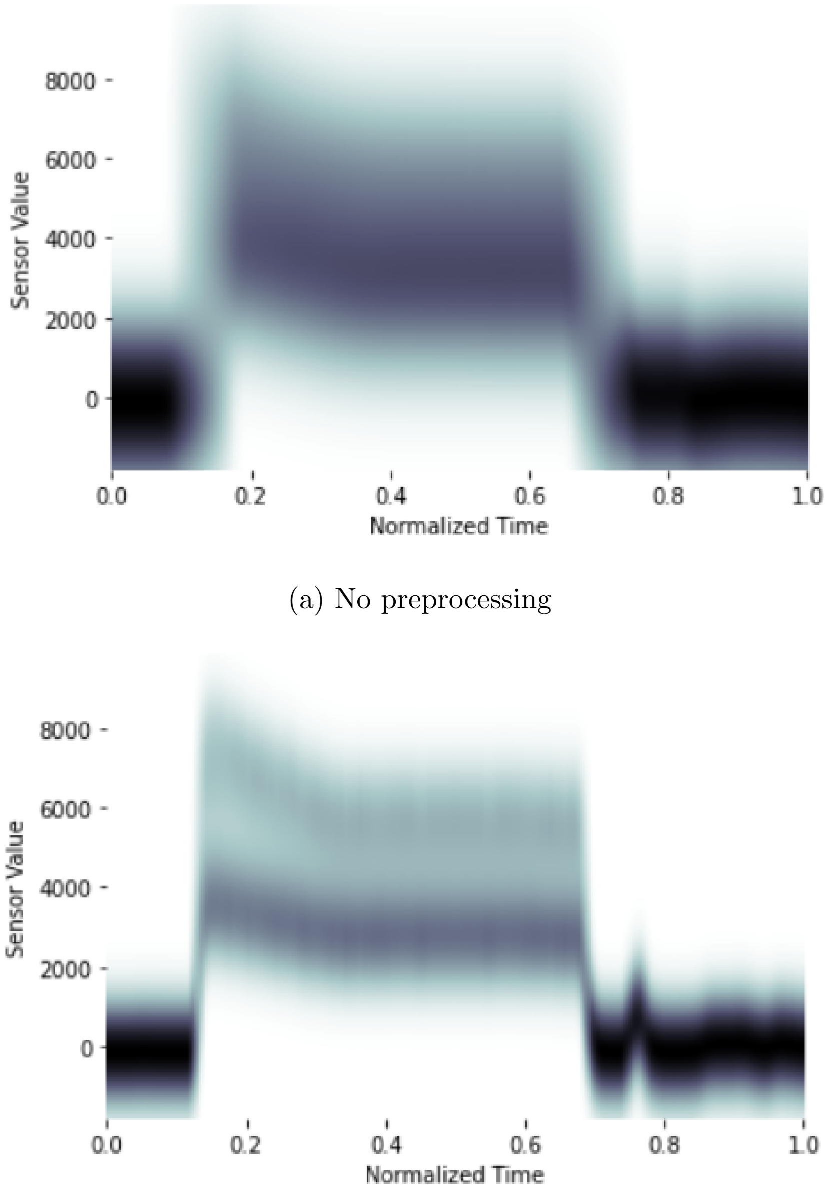
The increase in performance in the KDE model is likely due to the improved probability distributions that are able to be constructed. An example of one of the sensor distributions is visualized in Figure 4.10. When no preprocessing is used, the distribution is blurry, with wide allowances for signals. However, using the step-constrained DTW as a preprocessing method aligns the signals in time much more precisely, thus allowing the KDE model to build a more accurate distribution. The visual in Figure 4.10b is therefore much sharper, with narrower regions of high probability. Therefore, the model is more sensitive when signals stray from these regions, which improves anomaly detection.



(a)

(b)

Figure 4.9: An example of the different cost matrices for an anomalous signal that is detected using the step constrained DTW, but is missed using the Sakoe-Chiba band DTW. A noticeable difference between the two warp paths is a singularity in the bottom-left of the Sakoe-Chiba cost matrix.



(a) No preprocessing

(b) With DTW alignment

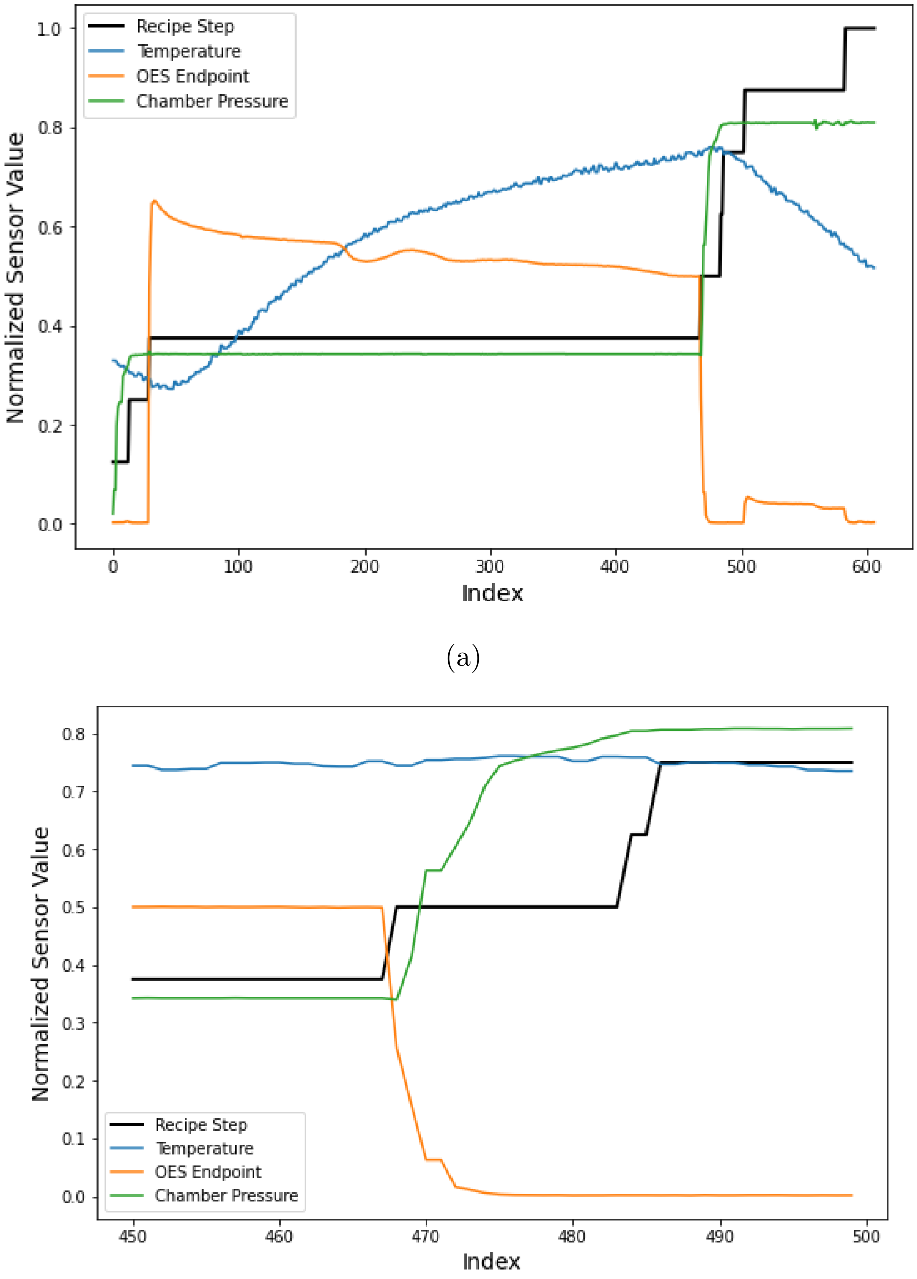
Figure 4.10: The use of step-constrained DTW as a preprocessing alignment method improves the quality of the calculated kernel density estimate probability distributions.

4.5 Effect of Signal Lead/Lag

As discussed in previous sections, the time at which the step transition occurs can potentially lead or lag any features that occur in the sensor signals. The amount of this delay is inherent in the industrial dataset, but can be controlled in the synthetic dataset. By injecting differing amounts of delay, we can explore how the different methods respond as the signals increasingly behave like real-world data with small amounts of delay.

An example of sensor signal timings for the plasma etch dataset is shown in Figure 4.11. The temperature, endpoint, and pressure sensors are chosen to display how different physical sensors respond as the recipe progresses. These sensor signals are overlaid on top of the recipe step signals. The bottom plot shows a zoomed in view of steps three through six, including the endpoint step, so that the delays are visible. We can see that the measured temperature moves slowly during the etch process, whereas the other parameters respond more quickly. When the main etch finishes and the transition between step three and four (at an index value of approximately 468) occurs, we see that the endpoint signal begins decreasing rapidly before the recipe step signal increments. This makes physical sense, as the rapid change in the endpoint signal is the indicator that the material property on the wafer has changed and the recipe should move to the next step. Then the chamber pressure begins to increase not long after the recipe step signal increments. This is likely in response to machine settings changing during the new step. The lead and lag nature of these signals is significant, and is explored further using the synthetic dataset.

Each run in the synthetic dataset has three recipe steps, the lengths of which are



(a)

(b)

Figure 4.11: Sensor and recipe step signals from one run in the plasma etch dataset. Sensor signals can both lead and lag the recipe step signal. (a) Overview of the full run. (b) Zoomed in view of steps 3-6.

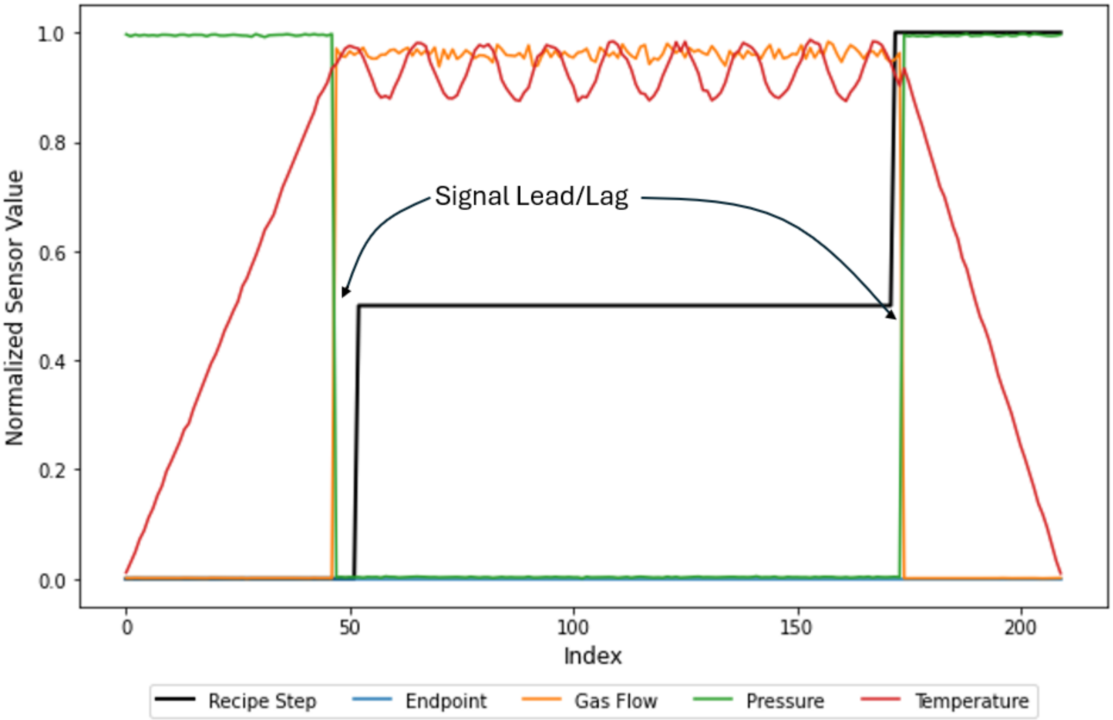
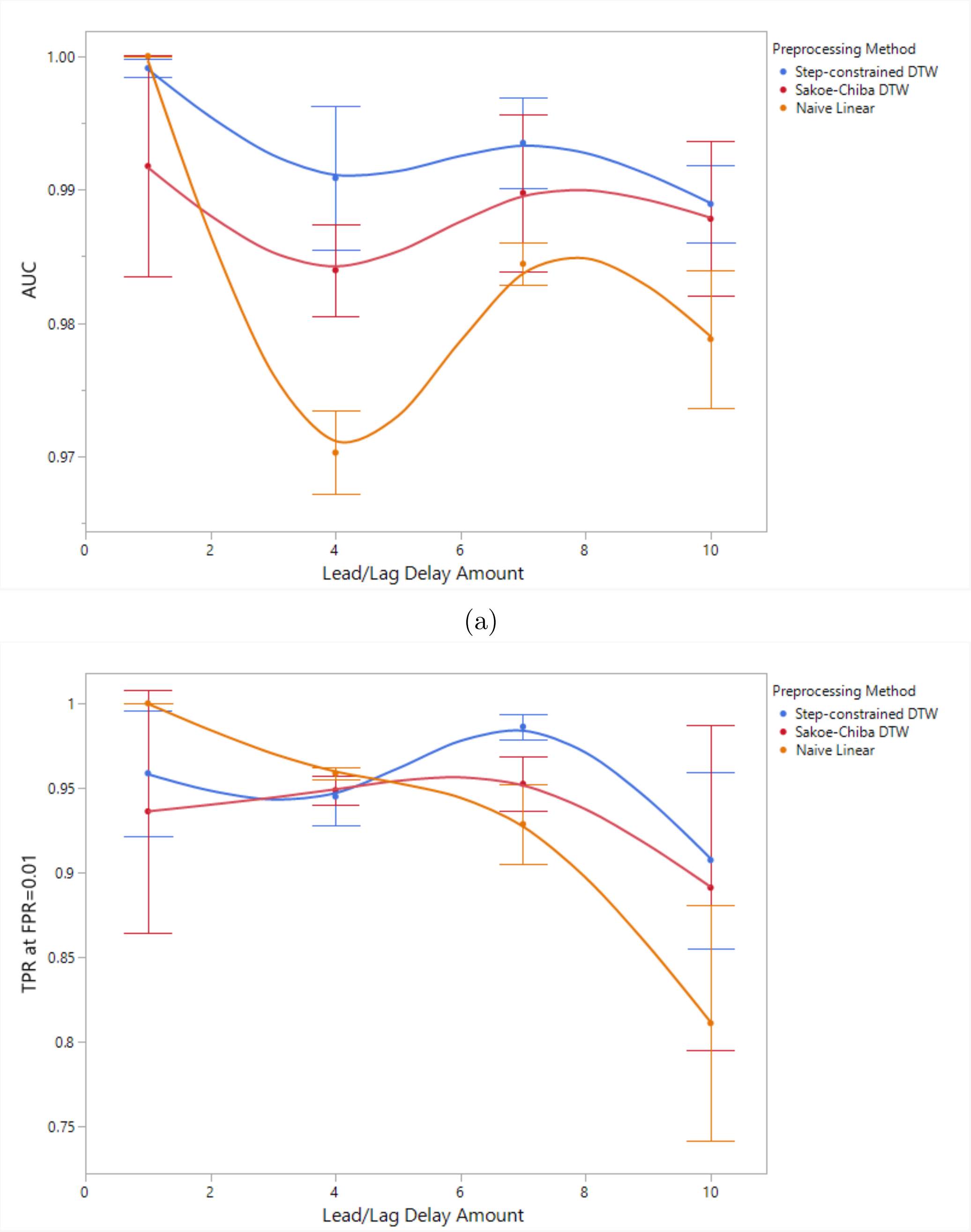


Figure 4.12: The recipe step signal transitions in the synthetic dataset are made to lead or lag features in the sensor signals.

normally distributed. For this experiment, a small amount of positive and negative delay is added to each step to offset it from the recipe step signal. This lead and lag signal delay is shown in Figure 4.12. All of the physical sensor signals still respond at the same time, but the addition of delay offsets this response slightly from the time at which the recipe step transitions.

For this experiment, the mean value of the delay is increased gradually, using the values 1, 4, 7, and 10. A value of zero delay would correspo!pdfnd to the results discussed previously in Section 4.4. The results for both AUC and true positive rate (TPR) at a false positive rate of 1% are shown in Figure 4.13. The naive linear rescale method significantly degrades in both AUC and TPR as the delay increases. This supports the hypothesis that the linear naive method performs much worse on the plasma etch dataset due to the unavoidable delays in real processing data.



(a)

(b)

Figure 4.13: The performance of the naive linear method significantly degrades as the delay is increased. Error bars show one standard deviation.

Chapter 5

Applications to Clustering

Step-constrained DTW can also be used for applications other than signal alignment. This chapter explores clustering methods. The DTW algorithm is frequently used as a distance measure for time series clustering methods, as discussed in Section 2.5. Step-constrained DTW can also be applied in the same manner, and is demonstrated on the plasma etch dataset described in Section 4.1.2.

5.1 Hierarchical Clustering

We first present a small example of hierarchical clustering with a total of ten wafer runs, five nominal and five anomalous. An agglomerative, or “bottom-up,” clustering method is applied. Each sample begins as a single cluster. The distance between these clusters is defined using a linkage criterion, which depends on a distance function. For our testing, we use single linkage and the distance function is fulfilled using the step-constrained DTW distance measure. At each iteration, the two closest clusters are merged together, and the distances are recalculated. This process continues until all the clusters have been merged, or an alternate stopping criterion is met.

The recipe step signals are employed for developing the constraints, and the same five relevant sensors are used in the distance calculation. The endpoint signals are displayed next to the dendrogram grouping in Figure 5.1. The dendrogram is a tree-like diagram that displays the cluster arrangements at each iteration. The first split correctly separates all of the anomalous runs from the nominal runs.

5.2 Density Clustering

Step-constrained DTW is also applied to density-based spatial clustering of applications with noise (DBSCAN). DBSCAN is a popular clustering method that is particularly effective for datasets with noisy or arbitrarily-shaped clusters. Unlike other unsupervised clustering methods, it does not require the user to predetermine the number of clusters.

As implied in the name, DBSCAN operates on the idea of density. Clusters are defined as dense regions with many data points, separated by sparse areas with low density. Each data point is classified as one of three types: a core point, a border point, or a noise point. Core points are located at the center of the cluster, while border points are found near the periphery. Noise points, also called outliers, are located further from the cluster. DBSCAN has two hyperparameters, $MinPts$ and eps, that must be set. The distance eps is the maximum distance between two points in order for them to be considered neighbors. Then $MinPts$ refers to the minimum

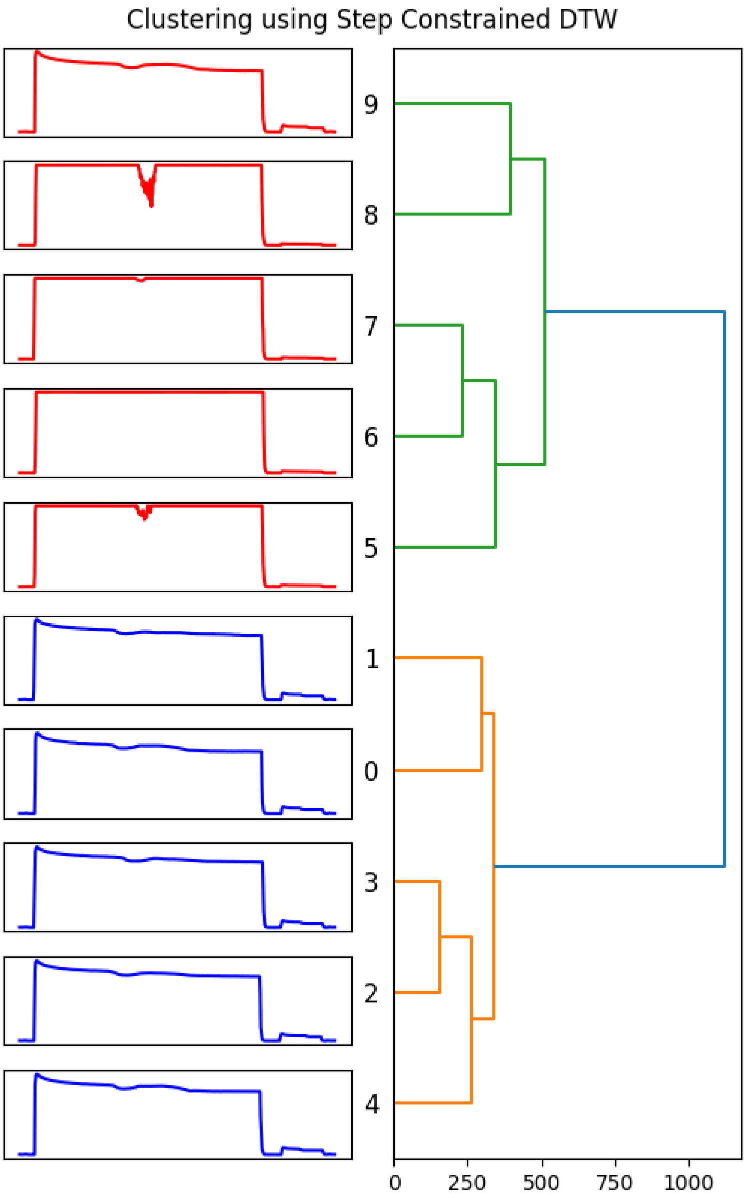


Figure 5.1: Hierarchical clustering of 10 plasma etch samples, containing 5 nominal runs (blue) and 5 anomalous runs (red). Using the step-constrained DTW distance measure results in accurate separation into two clusters (anomalous, nominal) at the top level.

number of points that must be within radius *eps* in order for a point to be considered a core point.

In order to select the *minPts* parameter for the plasma etch application, the rule of thumb of $minPts \geq D + 1$ is used, where D is the number of dimensions in the dataset. Since five recorded sensors are used for the distance calculations, *minPts* is set to six. Then *eps* is set to 100 after trial and error. For this experiment, 50 good wafer runs and 50 anomalous wafer runs are randomly selected from the dataset. The sensor signals are z-score normalized prior to the step-constrained DTW distance calculations.

The DBSCAN clustering finds three main clusters in the testing set. These clusters are visualized in Fig. 5.2 in both the temperature and optical endpoint signals. At first consideration, two clusters, one each for nominal and anomalous, might make sense. However, since there are different anomalous shapes in this dataset, a higher number of clusters yields better results. All of the samples in Cluster 1 are nominal wafer runs, while Clusters 2 and 3 contain only anomalous runs. The different anomalous shapes are particularly evident when viewing the anomalous optical endpoint signals. All of the purple runs belonging to Cluster 3 have strikingly high values during the etch, whereas the orange signals of Cluster 2 have more typical values. However, these Cluster 2 signals still have an anomalous shape in the temperature plot.

These examples demonstrate that step-constrained DTW can be used as a distance measure for various clustering applications. This can be useful for classification and monitoring of semiconductor processing sensor signals.

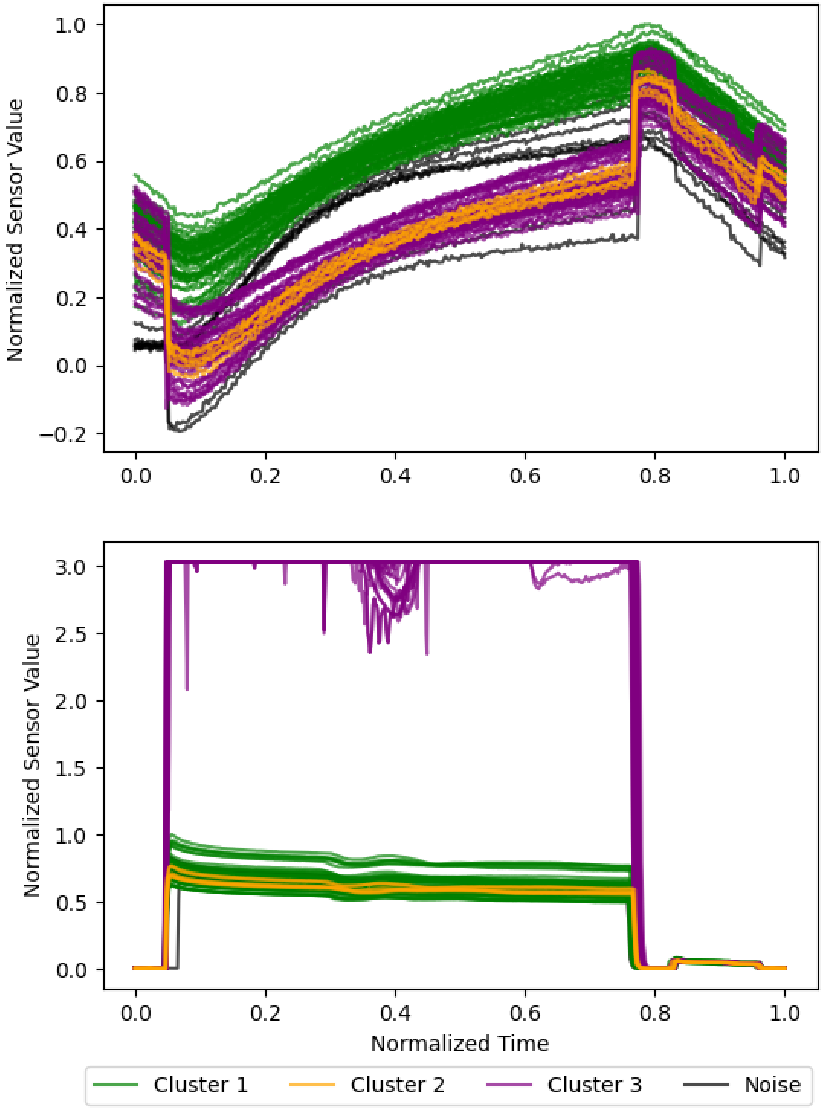


Figure 5.2: Temperature (top) and optical endpoint (bottom) sensor signals colored by the cluster label found using step-constrained DTW as the distance metric for the DBSCAN algorithm.

Chapter 6

Conclusions

In this thesis, a constraint method for the dynamic time warping algorithm is proposed. This method utilizes knowledge of semiconductor processing recipe step transitions to improve the signal alignments. Reference signals are generated using DTW barycenter averaging, which is established to be preferred over soft-DTW due to the improved representation and faster computation time. Experimental results on both synthetic and industrial datasets show that the step constraints outperform other preprocessing alignment methods. Step-constrained DTW is also demonstrated to be a useful distance measure for two popular clustering methods. These constraints are flexible enough to be applied to other processing datasets that contain recipe step signals.

In future work, further applications of step-constrained DTW may be explored. Anomaly detection methods could utilize the DTW distance measure or signal mappings. Additional work could also move beyond anomaly detection to anomaly classification, such as using k - nearest neighbor clustering with step-constrained DTW as the distance metric.

Bibliography

[1] Muhammad Zeeshan Arshad, Javeria Muhammad Nawaz, and Sang Jeen Hong. “Fault Detection in the Semiconductor Etch Process Using the Seasonal Autoregressive Integrated Moving Average Modeling”. In: *Journal of Information Processing Systems* 10.3 (Sept. 2014), pp. 429–442.

[2] Donald J Berndt and James Clifford. “Using Dynamic Time Warping to Find Patterns in Time Series”. In: *AAA1-94 Workshop on Knowledge Discovery in Databases* (1994).

[3] K. Selçuk Candan et al. “sDTW: Computing DTW Distances Using Locally Relevant Constraints Based on Salient Feature Alignments”. In: *Proceedings of the VLDB Endowment* 5.11 (July 2012), pp. 1519–1530.

[4] Marco Cuturi and Mathieu Blondel. “Soft-DTW: A Differentiable Loss Function for Time-Series”. In: *International Conference on Machine Learning* (2017).

[5] Chenxu Dai, Kaibo Wang, and Ran Jin. “Monitoring Profile Trajectories with Dynamic Time Warping Alignment”. In: *Quality and Reliability Engineering International* 30.6 (2014), pp. 815–827.

[6] Hoang Anh Dau et al. “Optimizing Dynamic Time Warping’s Window Width for Time Series Data Mining Applications”. In: *Data Mining and Knowledge Discovery* 32.4 (July 2018), pp. 1074–1120.

[7] Diab Mahmoud Diab et al. “Anomaly Detection Using Dynamic Time Warping”. In: Aug. 2019, pp. 193–198.

[8] Jianshe Feng et al. “Trace Abstraction: A Novel Method to Enhance Fault Detection in Semiconductor Manufacturing Processes with An Optimization Approach”. In: *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*. Oct. 2021.

[9] Zoltan Geler et al. “Dynamic Time Warping: Itakura vs Sakoe-Chiba”. In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. July 2019.

[10] Toni Giorgino. "Computing and Visualizing Dynamic Time Warping Alignments in *R*: The Dtw Package". In: *Journal of Statistical Software* 31.7 (2009).

[11] Han He. "Applications of Reference Cycle Building and K-shape Clustering for Anomaly Detection in the Semiconductor Manufacturing Process". Master's Thesis. Massachusetts Institute of Technology, 2018.

[12] F. Itakura. "Minimum Prediction Residual Principle Applied to Speech Recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1 (Feb. 1975), pp. 67–72.

[13] Brijnesh J. Jain. *Semi-Metrification of the Dynamic Time Warping Distance*. Sept. 2018. arXiv: `1808.09964 [cs, stat]`.

[14] Michael Jones et al. "Exemplar Learning for Extremely Efficient Anomaly Detection in Real-Valued Time Series". In: *Data Mining and Knowledge Discovery* 30.6 (Nov. 2016), pp. 1427–1454.

[15] Eamonn Keogh and Jessica Lin. "Clustering of Time Series Subsequences Is Meaningless: Implications for Previous and Future Research". In: *Third IEEE International Conference on Data Mining* (2003).

[16] Eamonn J. Keogh and Michael J. Pazzani. "Derivative Dynamic Time Warping". In: *Proceedings of the 2001 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2001, pp. 1–11.

[17] Matej Kloska, Gabriela Grmanova, and Viera Rozinajova. "Expert Enhanced Dynamic Time Warping Based Anomaly Detection". In: *Expert Systems with Applications* 225 (Sept. 2023), p. 120030. arXiv: `2310.02280 [cs]`.

[18] Christopher I. Lang et al. "One Class Process Anomaly Detection Using Kernel Density Estimation Methods". In: *IEEE Transactions on Semiconductor Manufacturing* 35.3 (Aug. 2022), pp. 457–469.

[19] Gyeong Taek Lee and Kangjin Kim. "Abnormal Chamber Detection in the Etching Process Using Time-Series Data Augmentation and Soft Labeling". In: *IEEE Sensors Journal* 23.5 (Mar. 2023), pp. 5084–5093.

[20] Jason Lines and Anthony Bagnall. "Time Series Classification with Ensembles of Elastic Distance Measures". In: *Data Mining and Knowledge Discovery* 29.3 (May 2015), pp. 565–592.

[21] Oumaïma Makhlouk. "Time Series Data Analytics: Clustering-Based Anomaly Detection Techniques for Quality Control in Semiconductor Manufacturing". Master's Thesis. Massachusetts Institute of Technology, 2018.

[22] Mingyan Teng. “Anomaly Detection on Time Series”. In: *2010 IEEE International Conference on Progress in Informatics and Computing*. Shanghai, China: IEEE, Dec. 2010, pp. 603–608.

[23] Cory Myers, Lawrence P. Rabiner, and Aaron E. Rosenberg. “Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-28.6 (1980), pp. 623–635.

[24] Francois Petitjean, Alain Ketterlin, and Pierre Gancarski. “A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering”. In: *Pattern Recognition* 44 (Sept. 2010), pp. 678–693.

[25] François Petitjean et al. “Faster and More Accurate Classification of Time Series by Exploiting a Novel Dynamic Time Warping Averaging Algorithm”. In: *Knowledge and Information Systems* 47.1 (Apr. 2016), pp. 1–26.

[26] Lawrence P. Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[27] Chotirat Ann Ratanamahatana and Eamonn Keogh. “Making Time-series Classification More Accurate Using Learned Constraints”. In: *Proceedings of the 2004 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2004, pp. 11–22.

[28] H Sakoe and S. Chiba. “A Dynamic Programming Approach to Continuous Speech Recognition”. In: *International Congress on Acoustics* 3 (1971), pp. 65–68.

[29] H. Sakoe and S. Chiba. “Dynamic Programming Algorithm Optimization for Spoken Word Recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (Feb. 1978), pp. 43–49.

[30] Stan Salvador and Philip Chan. “FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space”. In: *Intelligent Data Analysis* 11.5 (Oct. 2007), p. 11.

[31] Enrique Vidal et al. “On the Verification of Triangle Inequality by Dynamic Time-Warping Dissimilarity Measures”. In: *Speech Communication* 7.1 (Mar. 1988), pp. 67–79.

[32] T. K. Vintsyuk. “Speech Discrimination by Dynamic Programming”. In: *Cybernetics* 4.1 (1968), pp. 52–57.

[33] Ashani Wickramasinghe et al. “An Anomaly Detection Method for Identifying Locations with Abnormal Behavior of Temperature in School Buildings”. In: *Scientific Reports* 13.1 (Dec. 2023), p. 22930.

[34] Wang Yong et al. “Anomaly Detection of Semiconductor Processing Data Based on DTW-LOF Algorithm”. In: *2022 China Semiconductor Technology International Conference (CSTIC)*. June 2022, pp. 1–3.

[35] Jidong Yuan et al. “A Large Margin Time Series Nearest Neighbour Classification under Locally Weighted Time Warps”. In: *Knowledge and Information Systems* 59.1 (Apr. 2019), pp. 117–135.

[36] Jiaping Zhao and Laurent Itti. *shapeDTW: Shape Dynamic Time Warping*. June 2016. arXiv: `1606.01601 [cs]`.