Anomaly Detection and Feature Alignment for Time Series Data

by
Kishansingh Rajput

A Thesis submitted to the Department of Computer Science,

College of Natural Sciences and Mathematics

in partial fulfillment of the requirements for the degree of

Master of Science
in Computer Science

Chair of Committee: Dr. Guoning Chen

Committee Member: Dr. Ricardo Vilalta

Committee Member: Dr. Matthew Nicol

University of Houston
May 2020

ACKNOWLEDGMENTS

I received a great deal of support and assistance with all of the writing and thesis tasks. I would like to first thank my advisor Dr. Guoning Chen for his invaluable expertise, especially in formulating a research topic and methodology. He is one of the smartest people I have met. We had very insightful discussions about research as well as recent technologies. Collectively, these discussions and his guidance improved my confidence and helped me in my research.

I am also thankful to Dr. Ricardo Vilalta for his continuous support and guidance during my entire Master's program and his valuable time as my guide and research advisor. I would also like to thank Dr. Matthew Nicol for his support and time.

It has been great working with my fellow researchers: Michael, Lieyu, Marzieh, Duong, Naeem, Phang, Dana, and Zhenyu.

I am grateful to my friends Kevin, and Jinali for their continuous support during the course of my thesis work.

I am proud of my parents, the way they had brought me up, they are my role models. I will never be able to thank them enough for their support, and unconditional love and care. They always wanted and delivered best possible opportunities for their children. My younger brother, Narayan, also deserves my wholehearted thanks for his support. Last but by no means least, is the recognition of support I received from my fiancée. Thanks to Durga for always being by my side.

ABSTRACT

Time series data are stemming from various applications that describe certain observations or quantities of interest over time. Their analysis typically involves the comparison (with reference data for anomaly detection) and feature alignment across different time series data sequences. A general technique for anomaly detection via visualization is to compare a live signal along with reference sequences. Currently, the standard methods used in the industry are line/scatter plots. Due to limitations such as cluttering, lack of quantitative information etc., these plots are not effective. In this thesis, a probabilistic envelope based technique is proposed for the visualization and anomaly detection of time series data. This technique provides quantitative information, is able to avoid the outliers in the reference data, and works well even with a large number of reference sequences. To demonstrate the practical use of the probabilistic envelope technique, it is applied to the detection of over/under gauge of bore holes (wells). The implementation of gauge detection along with some results is also presented in this thesis. For feature alignment, the Dynamic Time Warping (DTW) is the standard approach to achieve an optimal alignment between two temporal signals. There are different variations of DTW proposed to address different needs of signal alignment or classifications. However, there is no a comprehensive evaluation of their performance in these time series data processing tasks. Most DTW metrics are reported with good performance on certain types of time series data without a clear explanation of this performance. To address that, a synthesis framework is proposed to model the variation between two time series data sequences for comparison. The synthesis framework can produce a realistic initial signal and deform it with controllable variation that mimics the real-world scenarios. With this synthesis framework, a large number of time series pairs with different but known variations can be produced, which are used to assess the performance of a number of well-known DTW measure in the tasks of alignment and classification. Their performance on different types of variations are reported and the proper DTW measure is suggested based on the type of variations between two time series sequences. This is the first time such a guideline for selecting proper DTW measure is presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Time series data are an important class of temporal data, that is a collection of observations made in chronological order. Unlike traditional data sets, time series data are not considered as individual numerical data fields, rather they need to be considered as whole. Time series data have an explicit dependence of data points as they are arranged on time axis. It is typically assumed that these data points are generated at a uniform time interval, but it is not always true. Time series whose data points are generated at a uniform time interval are called regular time series, and others are called irregular time series. The physical phenomena may be continuous/regular, but the observations may not be taken at regular time interval, and hence it results in irregular time series.

Another broad classification of time series data is based on the number of variables. If a time series has only one variable changing over time, it is called a **uni-variate time series**. If more than one variable are changing over time then it is called a **bivariate, trivariate, or multivariate time series**. Time series data can also be classified based on their nature. A **deterministic time series** is one which can be expressed explicitly by a mathematical expression. Its nature is not random, or it does not have any probabilistic nature. If the time series cannot be expressed as a mathematical formula and has random nature, then it is called a **non deterministic time series**. A time series is called **stationary** if its statistical properties such as mean, variance etc. do not depend on time. If the statistical properties change over time then it is called a **non stationary time series**.

1.1 Problem Statements

Analysis of time series data can provide very useful insights into many applications, like finance, engineering, biology, etc.. In the time series data analysis, there are mainly four categories including pattern discovery (trends), clustering, classification, and rule discovery [8]. Although there are many visualization techniques available as a result of research in this direction, there are no generalized approach or guidelines to all the time series data. Often, people need to create their

own visualization techniques depending on their data and the desired outcome.

One such problem is to visualize multiple time series sequences together. In many applications experts rely on the real time data (either recorded by sensors or humans) to decide on the further actions to be taken at any moment in time. For example, in oil and gas industry, experts continuously monitor different logs while drilling, and insights from these time series sequences help them decide further actions. In such applications experts mainly look for patterns/trends. Experts may also be watchful for any anomaly or places where certain constraints are not followed by the observations, or certain statistical properties are followed by the time series. Timely decision making is sought, otherwise, one may end up wasting a large amount of money and manpower in crucial applications. In such scenarios where timely decision making is critical, effective and efficient visualization techniques are needed to detect the anomaly, trend deviation or certain parameter matrix. In many such applications experts have some reference signals and they want to visualize real time signal with the reference signals to detect the anomaly. Simple visualization techniques, such as line plots and scatter plots, are not effective, since they have many limitations, such as cluttering and occlusion.

Another very famous problem in time series data analysis is the alignment of two or more temporal signals. In many applications these temporal signals are needed to be aligned with one another, where one or more signals are considered as reference(s) and newly arriving signals are aligned with them. The dynamic time warping (DTW) is used to find an optimal alignment between two temporal signals. It tries to warp one signal over another non-linearly by stretching or shrinking it along its time axis. There is a distance measure, which is minimized by DTW while producing such warping. This warping is then used to align two signals. DTW has become a very popular measure in pattern matching for time-series data analysis and visualization [5, 31]. It also leads to a number of variations (e.g., derivative DTW, weighted DTW, etc.) to address specific needs in different applications [21, 19, 14, 30].

Despite its wide application, there lacks a general guidance on the selection of the proper DTW method for the needs of different applications, which requires a study on how different characteristics

of the two to-be-aligned signals affect the performance of different DTW methods. To investigate how different characteristics of the signals affect the DTW alignments, two signals with known alignment and known characteristics (or difference) should be given, which is difficult to satisfy with the signals from the real-world applications.

My thesis work aims to address the above challenges and makes the following contributions.

1.2 My Contributions

To address the first problem of visualizing multiple temporal signals together and detecting anomalies, a set of visualization techniques are developed and introduced in this thesis. In particular, a probabilistic envelope based visualization technique is developed, which provides control over different level of details and focus on new observations to effectively compare trends of the real time signal with those of multiple reference signals to quickly detect the anomalies. The envelope like plots are created from the reference signals with the help of probability theory and the real time temporal signal is being visualized over the envelopes.

This newly developed visualization technique can also be applied to multi-variate time series data when the interests are similar to what was described above. For example, it can be used to visualize structure of a well bore as well as over/under gauge if any. In what follows, I briefly introduce the application of well bore.

Drilling is a process of cutting through rock (and other) formations. Once a well is drilled, engineers put a casing (tubing) inside drilled well to protect and support the well-stream. Wear occurs in the casing due to many factors. It is a critical problem in oil wells, it can be disastrous. If not treated in time, it may lead to severe damage and ultimately abandonment of bore hole, causing huge loss. The probabilistic envelope based visualization technique can be used to visualize and detect over/under gauge with the help of Caliper log data. Caliper is a circular device with many fingers in all the directions, when it is run down the borehole, it provides measurements of radius through its fingers' movement. The probabilistic envelope based technique, with slight modification can be used to visualize the overall structure of the well. It can detect the over/under

gauge as well as reveal the qualitative analysis of the well compared to the ideal situation.

To address the second problem of time series alignment, a new synthesis framework to generate pairs of time series sequences with known and controllable variations is introduced. The synthesis framework consists of two steps: (1) generate as realistic as possible a time-series data and (2) produce a second time-series data with the known difference from the first one. To simplify the second step, only phase shifting/scaling and random peak insertions (or removal) are considered to variate the input time-series data in order to generate the second series. It is also shown in later sections that these two simple variations can be combined to generate signals with vastly different characteristics.

I apply the pairs of time-series data generated with the proposed synthesis framework with different but known variations to assess the performance of different DTW methods. To measure the quality of the alignment results with different DTW methods, a proper distance measure is needed. The distance measure used by most of the researchers is the aggregate distance between the magnitude of matched points, since the ground truth matching of the two signals over the time axis is usually unknown. I refer to this measure as the aggregate distance over magnitude (ADM). In the experiments, since I use synthesis framework to generate realistic time series sequence pairs, the correct matching of these signals over time axis is known. Thus, I introduce another matching evaluation measure, which I refer to as the Aggregate Distance Over Time (ADT) that aggregates difference of time value of matching points. For both ADM and ADT, the smaller their values are, the better the alignment is. With these two measures, I report the performance of different DTW methods when applied to pairs of time-series data with known types of variations.

I extend the evaluation to the real life data sets of streamlines where points on two streamlines are matched with one another, and very important data sets of gamma ray logs used extensively in oil and gas industries to find out the depth of different surface formation transitions under the earth crust. This work fills the gap between the extensive research on inventing different variations of DTW and their practical use cases. To support the evaluation on the real-world data, I first consider one of the two to-be-compared time-series data as the reference, then perform a fitting to

obtain the set of parameters that can be used to generate a synthetic time-series from the reference that is as close to the second time-series data as possible. We then apply different DTW methods to the pairs of the two time-series data and compare the results with the ones obtained by applying DTW on the alignment between the reference and the synthetic one.

The rest of my thesis is structured as follows. I provide a briefly review of the related work in Chapter 2. In Chapter 3, I will describe the probabilistic envelope based technique, its usage, algorithm and effectiveness. In Chapter 4, I will discuss applications of the envelope based technique in the visualization and detection of over/under gauge in oil wells. In Chapter 5, I will describe our new synthesis framework to generate realistic temporal signal pairs, I will also evaluate different variants of Dynamic Time Warping (DTW) with different variations of temporal signal pairs and present the results in Chapter 6. I will conclude my thesis work and discuss its limitations, and potential future work in this direction in Chapter 7.

2 Related Work

2.1 Visualization of 1D Time Series Data

There are many techniques proposed for different applications of time series data, such as cluster and calendar based visualization of time series data [32], visualizing time series data on spiral [33], and aggregation of data for effective visualization [20]. Many machine learning based methods are proposed for anomaly detection in time series data such as Long Short term networks [23], fuzzy c-mean clustering [16], and anomaly detection in ECG healthcare data [6]. There are no ensemble based visualization techniques developed for anomaly detection via visualization.

The current approach to this type of problems is to simply use line plots or scatter plots to visualize multiple signals together. An illustration of this approach is shown in fig:LineScatter. Comparison of one signal with another signal can be easily visualized using line plots, but for the applications where we need to compare one (current) signal with a large number of other (reference) signals, simple line plot or scatter plot is not effective due to the following reasons:

- They create clutters when the number of signals is large.
- They do not provide any quantitative information.
- They are difficult to interpret.
- They do not highlight the anomalies which are important for the decision making.

To overcome the above limitations, the probabilistic envelope based visualization and anomaly detection technique is proposed in the later sections.

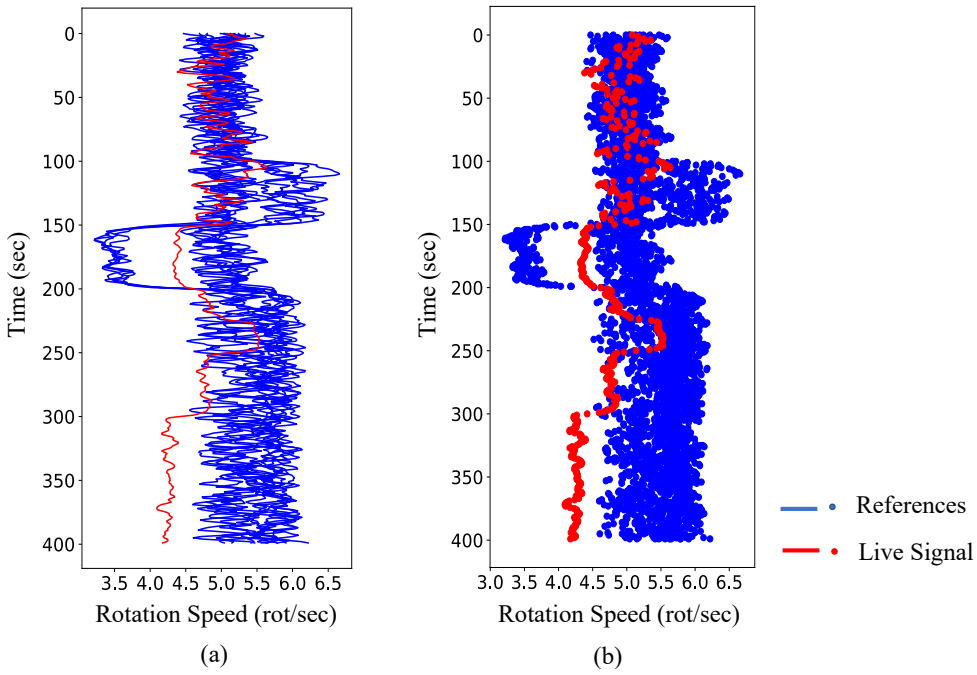

Figure 1: Visualization of rotational speed of drill bit along with reference data. Existing methods for visualization of anomaly: (a) Line plot and (b) Scatter plot.

2.2 Different Variants of Dynamic Time Warping

A large amount of time series data are being generated every day in many fields, such as manufacturing, oil and gas industries, engineering, finance, medicine, natural science and biology, etc.. Due to this, many interesting applications of DTW are seen in data mining. For example, DTW is being applied to fuzzy clustering [17], clustering with global averaging method on DTW [27], clustering with hidden Markov model and DTW [26], WDTW for time series classification [19], support vector based algorithm with WDTW for time series classification [18], and motif discovery [25]. It has been extensively used in speech recognition [29, 12], handwriting recognition [1], gesture recognition [4], signature recognition [10], ECG signal pattern recognition [15], and many others.

As a result of increasing importance of time series classification in many fields, lots of variations are proposed for different applications. Standard DTW is used for image matching by Rath and

Manmatha [28] and its performance is compared with other popular techniques. Gullo [13] proposed time series representational model, called derivative time series segment approximation. Jeong[19] applied weighted DTW (WDTW), derivative DTW (DDTW) along with weighted derivative DTW (WDDTW) on synthetic as well as real life datasets like Swedish leaf, lightening-2, ECG etc. and compared their performance with other techniques.

Keogh and Pazzani proposed derivative DTW (DDTW)[21]. Instead of considering the actual distance between the magnitudes of the time series pairs, they considered difference between derivatives. Their work lacks concreteness in terms of experiments since they used only three datasets. They did not test it in the context of the classification of time series. Later works, however, used DDTW and performed the evaluation [19, 11]. On the basis of this method, Kulbacki created a measure which took into account the standard distance between time series[22]. To combine Euclidean distance measure and estimated derivative distance, Kulbacki considered the product of these two.

Weighted dynamic time warping (WDTW) is being used for speech recognition [35]. It is also used for satellite image time series analysis [34], and has been extended to time-weighted DTW [24].

A Synthesis framework [9] has been introduced to distort a linear time series sequence. The main focus in [9] was on optimization of maximum allowed warping to constraint it, and considered only the scaling (of entire signal) as a distortion factor and no other variations.

Even after all these works, there are no proper guidelines to choose a variant of DTW measures for a specific problem domain or for the signals with particular characteristics. In this work, I try to fill this gap by evaluating these different variants of DTW measures over different characteristics of time series and present the results as a guideline.

3 Background

In this section, I provide a brief introduction to the two real-world applications where my proposed techniques will be applied to. I also briefly review some probability basis and existing DTW measures that will be needed for my later discussion.

3.1 Introduction to Bore-Hole Gauge

Wells are drilled by engineers to reach reservoirs at a certain depth in the earth's crust. It is a process of cutting through rock (and other) formations. The raw sides of the bore-hole cannot support themselves. If the drilled well is to become a production well, engineers put a casing (tubing) inside the drilled well to protect and support the well-stream, it is called completion process.

In the process of casing a well, steel pipes are run down the recently drilled bore-hole. This process is also called setting pipe. The space between the raw formation and the casing is filled with cement to attach the casing and make it stronger. In addition to providing stability and keeping the sides of the well from caving in the bore hole, casing protects the well-stream from outside contaminants, as well as any other reservoirs from the oil or gas that is being produced.

Casing wear (damage) occurs as a result of the drill string rubbing against the casing, high pressure, and temperature conditions. The wear depends upon the contact forces, the wear track length (distance one surface moved across the other), the nature of the surfaces in contact, the material strength and hardness, and the presence of third bodies or lubricants between the wearing surfaces. The main generator of wear track length is drill string rotation. In addition to this, bore-holes are not always vertical and sometimes they are inclined or even horizontal. In such wells there is more chance of wear at the bends.

fig:casingWear shows a typical example of casing wear (over/under gauge) in bore-hole. Casing wear is a critical problem in oil wells. If not treated on time, it may lead to severe damage and ultimately abandonment of bore hole, causing huge loss.

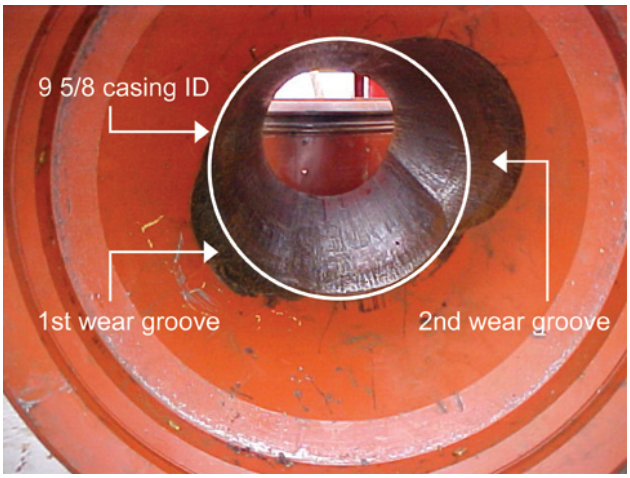

Figure 2: Casing Wear (Society of Petroleum Engineers website; URL: tinyurl.com/yav5jp7s)

3.2 Introduction to Detection of Surface Formation Transitions under Earth's Crust

While drilling, engineers need to continuously fine tune the parameters of drilling machines in order to efficiently cut through different formations. Different formations require different amount of mechanical forces for energy efficient and accurate drilling. If drilling engineers know the depths of the formation transitions beforehand, it makes the parameter tuning easy.

Depth wise gamma ray logs are recorded once wells are drilled. The depth of surface formation transitions under the earth can be found using Gamma ray logs, Resistivity logs and Spontaneous Potential logs. Geologists analyze these logs simultaneously to mark the boundaries of formation tops based on their domain knowledge. Among these three logs, gamma ray logs are the easiest to record and found in most of the wells while resistivity and potential logs are not always available. That said, geologists can mark the formation tops based on the gamma ray logs with/without any other additional information (logs). The process of matching the formation tops by geologists involve logs from one or more wells whose surface formations are known and logs from some other wells whose formations are not known, geologists try to match the depth level of formation tops from known logs (also called reference) to the logs whose formations are not known (also called

target) based on the trends/peaks of their respective gamma ray logs. Even experts cannot clearly explain what they are matching in those logs, they mostly look at the trends in logs and try to match similar trends across wells and depth levels.

This process can be automated with the help of 1D signal alignment algorithms. If reference gamma ray logs can be accurately aligned with the target gamma ray logs, surface formations transitions can be marked without any error by the software. This lowers the possibility of human errors which are mostly found in manual marking of these transitions by geologists. DTW is a good candidate algorithm for this type of alignment problems.

3.3 Probability Basics

Probability is a broad mathematical concept and its detailed discussion is beyond the scope of this thesis, but the basic notions of probability which are required to understand the content of this thesis are explained in this section.

Probability space: A probability space is a model based on three components, a sample space, an event set, and a probability distribution.

Sample space Ω: Consider an experiment, the set of all the possible outcomes are known. The set of all the possible outcomes is called the sample space (Ω). For example, each of the six outcomes in $\{1, 2, 3, ..., 6\}$ when tossing a dice.

Event set $\mathcal{F}$: It is all the subsets of Ω containing Ω that is closed under complementation and countable union. For example, in our earlier experiment of tossing a die, an event is "the die displays an odd number".

Probability distribution: It is a mapping from the set of all the events $\mathcal{F}$ to $[0, 1]$ such that $Probability[\Omega] = 1$ and for all mutually exclusive events $A_1.....A_n$,

$$Probability[A_1 \cup ..... \cup A_n] = \sum_{i=1}^{n} Probability[A_i]$$

In later sections (specifically sec-pel), the probability of an observation falling in certain magnitude range using the references will be computed. The assumption here, is that the current signal will follow the trend similar to the references so the proportion of the reference observations falling in certain range can be used to define probability of new observation falling in that range as described in fig:prob. The probability of a new observation falling in the magnitude range enclosed by the gray region in fig:prob is 0.6, since proportion of the reference observations falling in that range is 0.6.

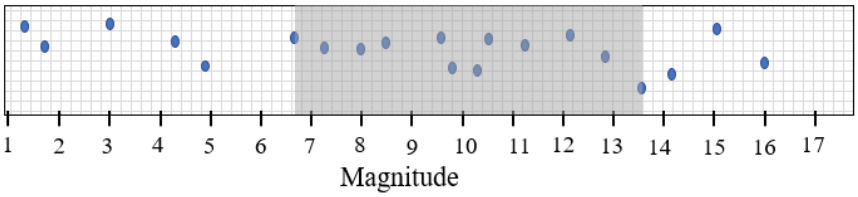

Figure 3: Probability of a new observation falling in certain magnitude range based on reference observations. A new observation has probability of 0.6 to fall in the magnitude range of $[6.6, 13.5]$ since it encloses 60% of the reference observations around the median.

3.4 Dynamic Time Warping and its Variants

Dynamic Time Warping (abbreviated as DTW)[2] is a widely used method to find warping/matching between two time series sequences. The methodology for DTW is as follows. Consider two time series sequences (or signals) $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_n\}$ of length m and n, respectively. A $m \times n$ matrix is created, where each entry (i, j) contains distance between x_i and y_j, such that $dist(i, j) = |x_i - y_j|$. In general, this distance is normalized to second normal form. The path from $(0, 0)$ to (m, n) in this matrix with minimum aggregate distance is selected. As shown in fig:dtwMethod the entry in each cell of the path matrix P(i, j) is the sum of dist(i,j) and min[P(i-1, j-1), P(i-1, j), P(i, j-1)] to find the path with minimum aggregate distance. This path is called the **warping path** [2]. This distance measure is based on ADM, in our evaluation we will show both ADM and ADT. The running time of DTW is quadratic since it uses Dynamic Programming, but many attempts are already made to reduce the runtime.

There are many variations of DTW already proposed that will be discussed later in this section.

One of the limitations of DTW is in the features it considers. It only considers Y-axis values of the time series; due to this it may not be able to accurately align series which are even slightly different (or shifting) on y-axis. One attempt to overcome this issue is the Derivative DTW (DDTW)[21]. DDTW is a variation of standard DTW where instead of taking raw series, derivative of each point is taken into consideration. It makes DTW more accurate in certain cases by considering direction information of the time series.

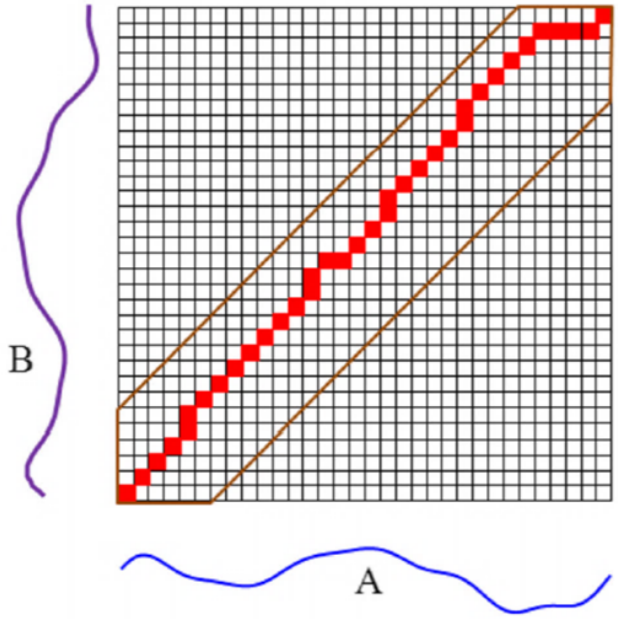

Figure 4: Warping path for alignment of two signals using DTW.

Another limitation of DTW is that it cannot accurately align two series when there is high variance, i.e., the magnitude of the matching points between two temporal signals are very different. When the warping path is too skewed it is more likely to have a large error. To overcome this issue, Weighted DTW (WDTW)[19] was introduced. WDTW weights each distance value in the warping matrix before they are considered for minimum distance path. High weight factor ensures the path to be diagonal and does not allow it to be skewed, while low weight factor allows more flexibility.

The weight factor is designed to be high around the central region of the series, since it is believed that signals are more stable around central region and low around the ends to accommodate high fluctuations. Weighted DTW inspired using Weights in DDTW, hence Weighted Derivative DTW (WDDTW) was also proposed in the same paper[19]. It is similar to WDTW, but weight factors are multiplied with differences of derivatives rather than differences of magnitude.

Constraint based approaches are used to restrict the skewness of the DTW warping path as well as to increase the runtime speed. The most famous and widely used approach is windowing[30] or lower bounding the distance. Windowing restricts the DTW warping path to remain only inside the window created in the distance matrix. The shape of this window can vary depending on the domain knowledge or fluctuations in the time series data. This approach helps improve both computation time and accuracy if the window is selected carefully.

Attempts are also made to use DTW in Machine Learning applications. DTW distance is being used as Neural Network node value [3] in classification of time series data, which is referred to as DTWNet. DTWNet is a simple neural network with one or more layers as DTW layer(s). Each DTW layer's nodes optimize the DTW distance value via back propagation. In another attempt DTW loss function is made differentiable in order to use it for regression. In SoftDTW [7] the DTW loss function is differentiable and hence allows computation of its value and gradient in quadratic time and space.

3.5 Simulated Annealing

Simulated annealing is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.

Simulated Annealing (SA) is an effective and general form of optimization. It is useful in finding global optima in the presence of large numbers of local optima. "Annealing" refers to an analogy with thermodynamics, specifically with the way that metals cool and anneal. Simulated annealing uses the objective function of an optimization problem instead of the energy of a material. The

algorithm is basically hill-climbing except instead of picking the best move, it picks a random move. If the selected move improves the solution, then it is always accepted. Otherwise, the algorithm makes the move anyway with some probability less than 1. The probability decreases exponentially with the "badness" of the move, which is the amount ΔE by which the solution is worsened.

$$Prob(accepting\_uphill\_move) = 1 - e^{(\Delta E/kT)}$$

A parameter T is also used to determine this probability. It is analogous to temperature in an annealing system. At higher values of T, uphill moves are more likely to occur. As T tends to zero, they become more and more unlikely, until the algorithm behaves more or less like hill-climbing. In a typical Simulated Annealing optimization, T starts high and is gradually decreased according to an "annealing schedule". The parameter k is some constant that relates temperature to energy (in nature it is Boltzmann's constant).

We will use Simulated Annealing to assess the effectiveness of our synthesis framework for 1D signals. We will try to find the parameters of our framework to transform one signal to another using Simulated Annealing.

4 Envelope based Visualization Technique

4.1 Probabilistic Envelope based Visualization

In this section, a novel probability based visualization technique for simultaneous visualization and monitoring of multiple 1D time series data is introduced. It is referred to as the Probabilistic Envelope based technique or PE-technique in short. This technique is very useful for *anomaly detection*. While visualizing multiple signals as reference and live/current signal as target, it can effectively highlight the regions where anomaly occurs.

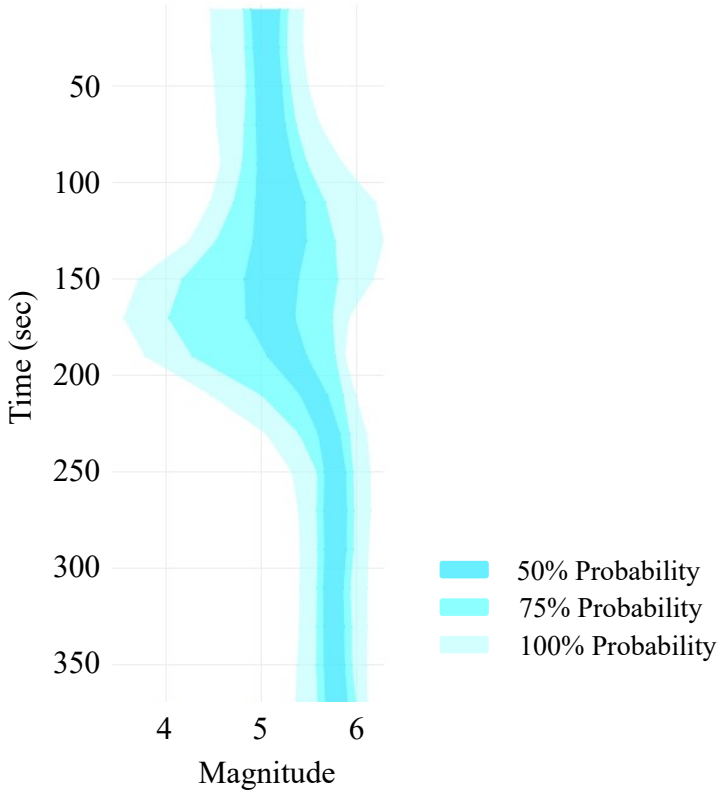

Figure 5: Illustration of the Probabilistic Envelope based Visualization technique. Number of layers and probability values assigned to each layer are user controllable. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

With this technique current signal is visualized over the likelihood distribution calculated using

the reference signals. The likelihood distribution resembles envelope structure with multiple layers as shown in fig:envelope2. The layers (shown with different color shading) in the envelope structure represents different likelihood of the reference signals, that is, how likely a certain percentage of the reference signals falling within the given layer. We estimate this likelihood by computing the proportion of points from the reference signals falling within the specific layer. Its algorithm is described in the next section. The user can decide these proportions based on their application.

4.1.1 Calculation of Probabilistic Envelopes

Input Parameters: List of reference signals, number of layers (m) along with their probability values. Step-size (optional) for clustering.

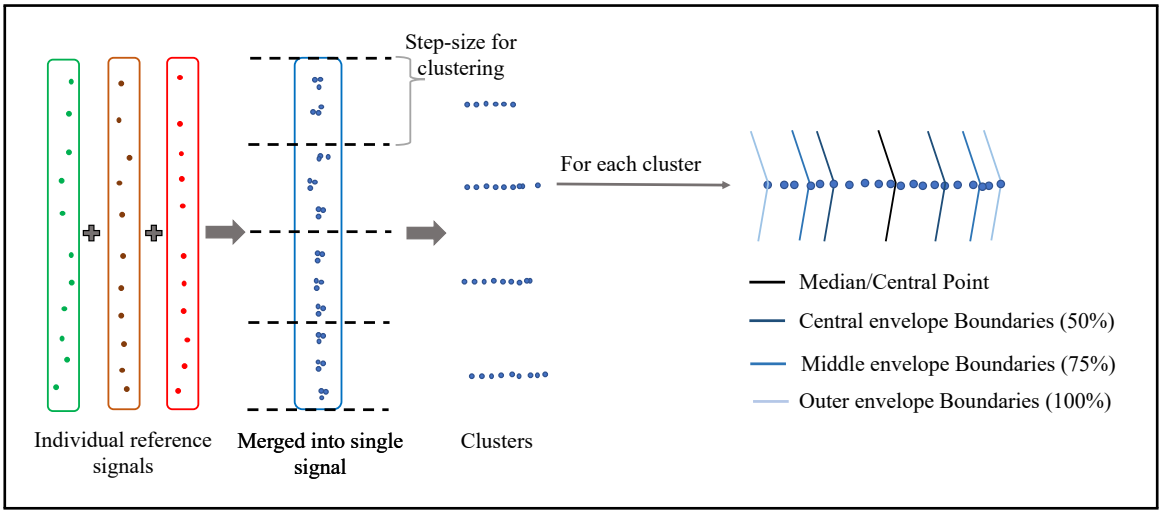

Figure 6: Pipeline to compute the probabilistic envelope boundaries.

Method: Firstly, all the reference signals are merged into a single temporal signal while preserving their time axis as shown in fig:envelope$_m ethod. Then, the observations are clustered using a step-size over time axis. Step-size can either be provided by the user or a default value can be used. For each cluster, likelihood$

For the visualization, darker shades for the envelopes with high probability values are used while gradually decreasing the darkness of the shades for the envelopes with lower probability values.

Algorithm 1 describes the implementation of envelopes computation. For a likelihood distribution with m layers, we need $2m$ boundaries. In the algorithm we called these boundary arrays as $boundary\_lower_i$ and $boundary\_upper_i$, representing lower and upper boundary of $envelope_i$, respectively.

Algorithm 1 Algorithm to calculate boundaries for envelope structure

Initialization : Initialize P with list of percentage values $P_1, P_2, ...P_m$ for all the m envelopes and D_0 as depth offset for clustering

From all the reference signals group signal points falling within each D_0 step (based on depth) as a cluster

for each cluster **do**

    Put all the points of the cluster to a list L

    Find Median M of the list L

    length ← length of L

    Initialize m temporary empty lists as $temp_1, temp_2, ...temp_m$

    for each P_i in P **do**

        Find number of points N_i in the envelope i using following formula

$$N_i = int(\frac{length \times P_i}{100})$$

        for j in range $1...N_i$ **do**

            X ← A point nearest to M in L

            Add X to $temp_i$

            Remove x from L

            Stop if L is empty

        append all the element of $temp_{i-1}$ to $temp_i$

    for i=1 to len(P) **do**

        Add $min(temp\_i)$ to $boundary\_lower_i$

        Add $max(temp\_i)$ to $boundary\_upper_i$

return $\{boundary\_lower_1, boundary\_upper_1, ...boundary\_lower_m, boundary\_upper_m\}$

4.1.2 Evaluation of Probabilistic Envelopes

As described in Algorithm alg:Algorithm$_E nvelopes, the computation of boundaries of the envelopes requires processi$
$time complexity of the above described algorithm is a function of r and n. The time complexity O(A) can be described as fo$
$O(r \times n) Note that the time complexity does not depend on the number of layers or the probability values assigned to each la$

$$O(A) = O\left(r \times \frac{T}{S}\right)$$

As the run-time complexity is dependent on the number of reference signals, the computation will be slow if the number of references is very large. Since the quality of the envelope structures are enhanced with increasing number of reference signals, there is a trade-off between accuracy and speed.

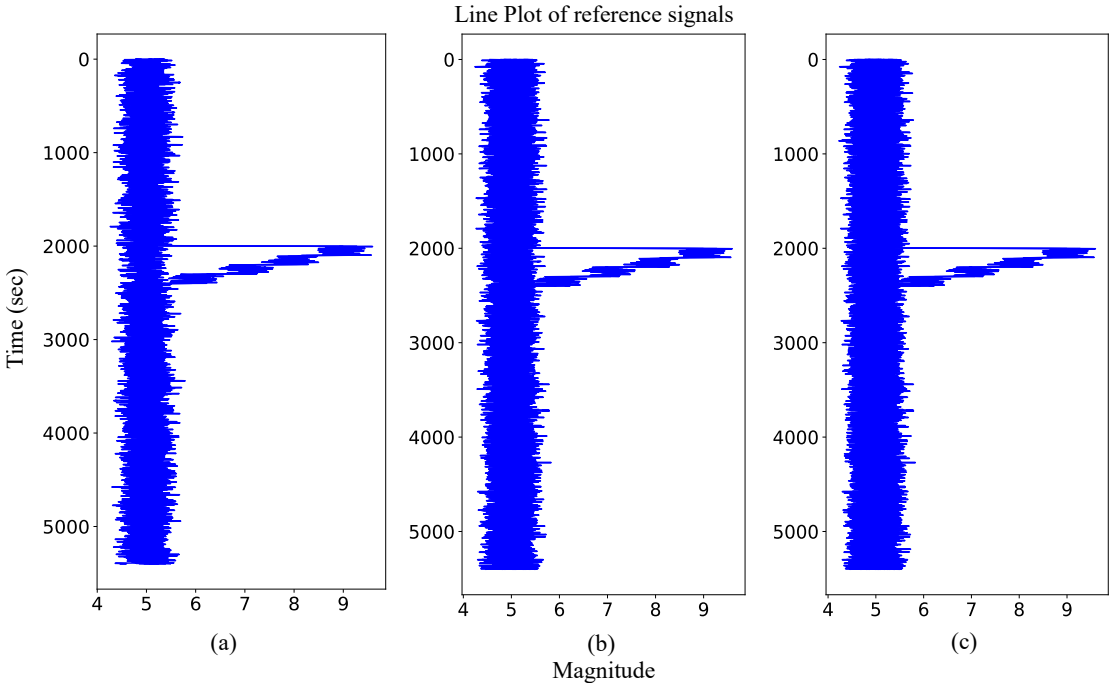

Figure 7: Standard method (line plots) for different number of reference signals with outliers. (a) 5 reference signals, 2 of them have outliers. (b) 10 reference signals, 3 of them have outliers. (c) 15 reference signals, 3 of them have outliers. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

While processing the live time series data, the envelope structure remains the same and does not require re-computation, hence once the likelihood (envelope structure) is computed, it can be used for infinitely long time unless there is a change required in the reference signals.

As mentioned earlier, this PE-technique can mitigate the effect of outliers. The standard method does not have significant difference on outlier visualization with small and large number of references as shown in fig:eval1. Our envelope technique can mitigate the effect of outliers with an appropriate number of layers and probability values, but this mitigation is more effective when the number of references is large, provided the majority of them do not have many outliers as shown in fig:eval2. Reducing the probability of outermost layer can avoid the outliers, but with a smaller number of references where the proportion of outliers is large, it needs a large decrease in probability value which affects the sections with no outliers. On the other hand, with a large number of references and a smaller proportion of outliers, it works really well with slight decrease in probability as can be understood from fig:eval2.

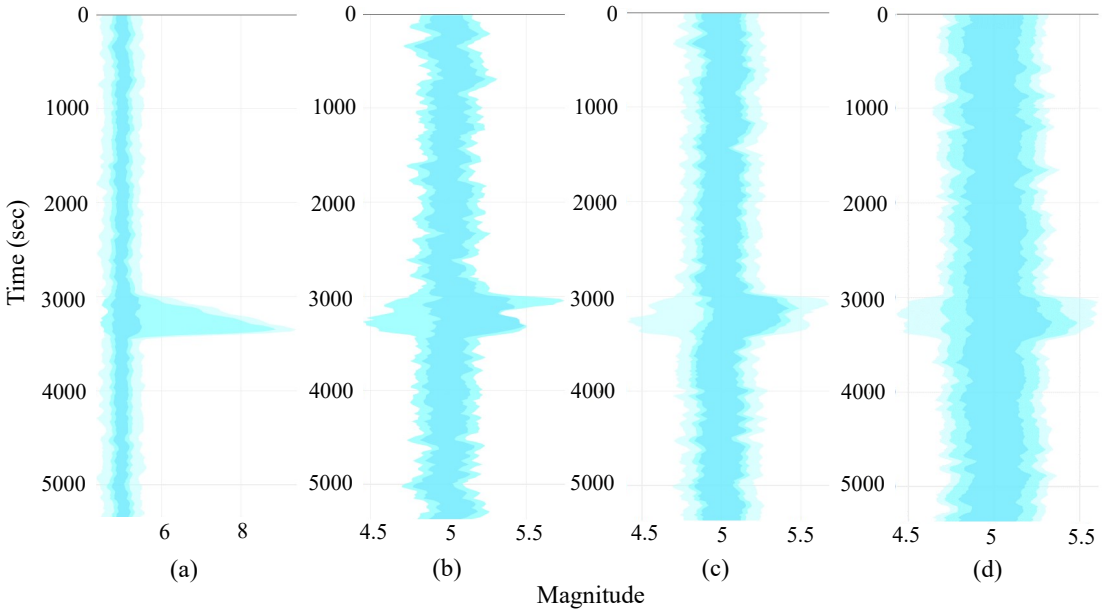

Figure 8: Envelope structures with different number of references and probability values to reduce the effect of the outliers. Some of the references have outliers in the time interval 3000-3500. (a) 5 reference signals, 2 with outliers, probability values used to construct the envelops are 50%, 70%, 100%, respectively. (b) References same as (a), two layers with probability 40% and 60%. (c) 10 reference signals, 3 of them have outliers, probability values for envelope construction are 50%, 60%, and 70%, respectively. (d) 15 reference signals, 3 of them have outliers, probability values are 50%, 70%, and 90%, respectively. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

4.1.3 Applications

As briefly described in sec-pel, the probabilistic envelope technique can be used for anomaly detection as well as noise reduction in references. It also provides user control over the precision in the anomaly detection. Users can control the probability values of the layers along with the threshold layer. In fig:envelope2 and fig:envelope3, the envelope structures consist of 3 layers with the probability values of 50%, 75% and 100%, respectively. Intuitively, the most inner (or central) layer of the envelope encloses 50% of the points/observations from reference signals, middle envelope layer

encloses 75% of the observations and the outer envelope layer encloses all the observations in the reference signals.

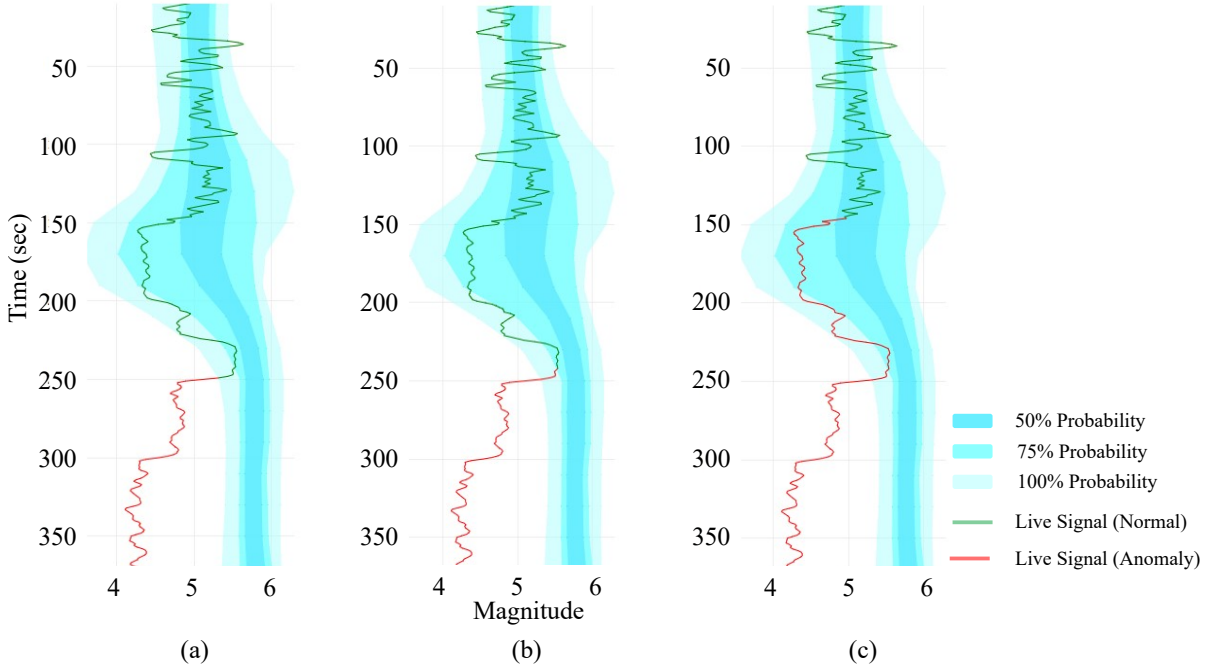

Figure 9: Probabilistic envelope based visualization for signal monitoring. Three probability layers, corresponding to 50%, 75%, and 100%, respectively, are used. The current signal is shown as the green line. The red sections of the current signal indicate places where the signal go outside of the specified likelihood threshold. (a) , (b), and (c) uses the outermost layer, middle layer, and central layer as threshold, respectively. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

To detect the anomaly, the user needs to mark a layer (based on probability values given initially for the computation of the envelopes) as threshold. Threshold layer is the outer most layer enclosing valid magnitude range for the current signal and all the observations falling outside of this envelope will be considered as anomaly. In fig:envelope3 (a) outer envelope (with 100% probability) is used as the threshold, hence all the line segments representing an observation outside this envelopes are marked with red color, while the normal observations falling within the outermost envelope

are shown with green color. Similarly, for fig:envelope3 (b) and (c), middle envelope (with 75% probability value) and central envelope (with 50% probability value), respectively, are used as threshold envelopes.

PE-technique can also be used to reduce the effect of outliers in the reference signals. As fig:reduceOutlierEffect shows, using different numbers of layers or probability values, the effect of outliers to the envelope can be mitigated. This can improve the efficiency and robustness of abnormal detection when the reference signals are noisy.

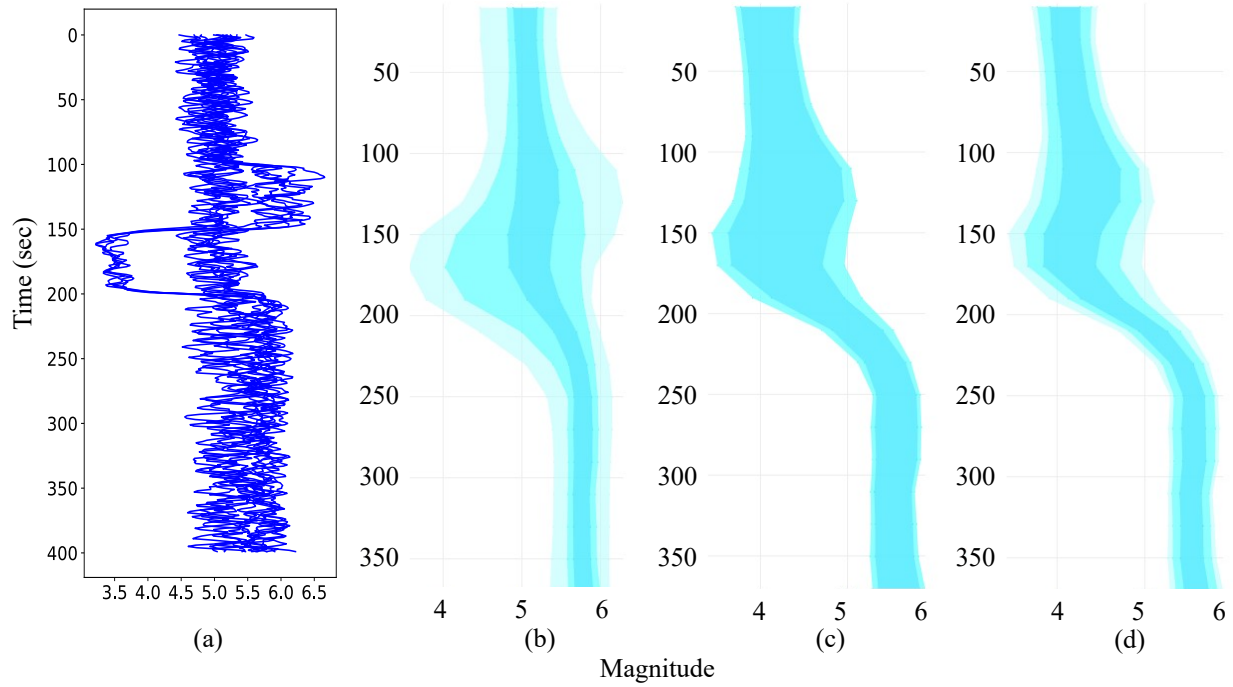

Figure 10: Reduce the effects of outliers by controlling number of layers and probability values. (a) Original line plot Method. (b) Probabilistic Envelope with 3 layers and (0.5, 0.75,1) Probabilities (c) probabilistic Envelopes with 2 Layers, (0.5, 0.7) Probability. (d) Probabilistic Envelopes with 3 layers, (0.3, 0.5, 0.7) probabilities. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

4.2 Rigid Boundary Based Visualization

In some applications, reference signals are not available or may not be useful. In such scenario, a rigid boundary based visualization and anomaly detection technique can be used. This can be easily accomplished using simple line plots with boundaries that follow certain function enclosing the normal value range. This function may be according to the seasonality (if any) or any other factor depending on the application, generally these boundaries are flat as shown in fig:hardboundary (a) with blue color.

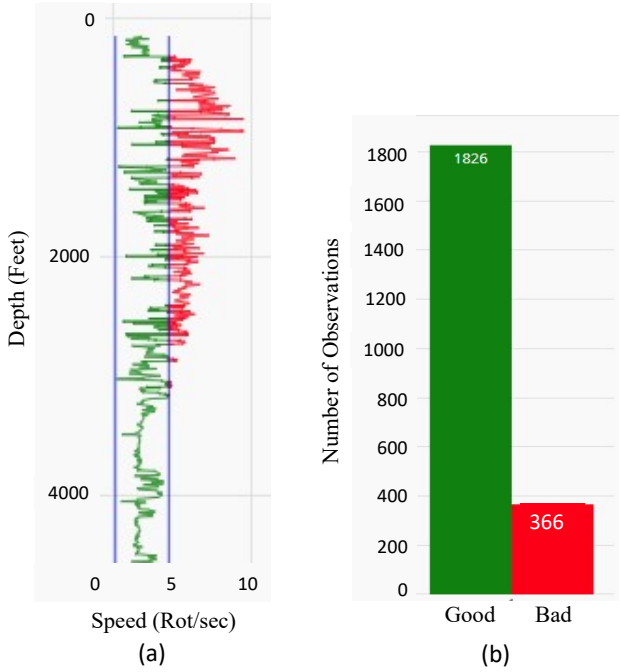

Figure 11: Hard boundary based signal monitoring.

The sections of the signal are marked with red color which has magnitude outside the boundaries. This technique provides quick detection of deviation or anomaly in the signal. This deviation may be due to various reasons, and the discussion of these errors is beyond the scope of this thesis.

To provide the quantitative measure of the consistency of the signal, results shown in fig:hardboundary

(a) are summarized in fig:hardboundary (b) in the form of histogram. Histogram shows the proportion of signal data points falling inside and outside the boundaries, respectively.

4.3 Rule Matching

In some applications, live incoming signal is required to be monitored in order to detect time interval where certain metric (rule) is matched. A simple line plot with shades as shown in fig:Focus, can provide an effective visualization in such applications. The metric can be any mathematical formula which is required to be followed by a section of the linear data; for example, a metric can be standard deviation higher than certain constant value or it can also be a complex mathematical formula. The simple shading method hides the section of time series data which does not follow the given metric while highlighting the sections where it does. In fig:Focus, section of the data where standard deviation is higher than 50 (for a window of 100 feet) are highlighted while hiding the other sections.

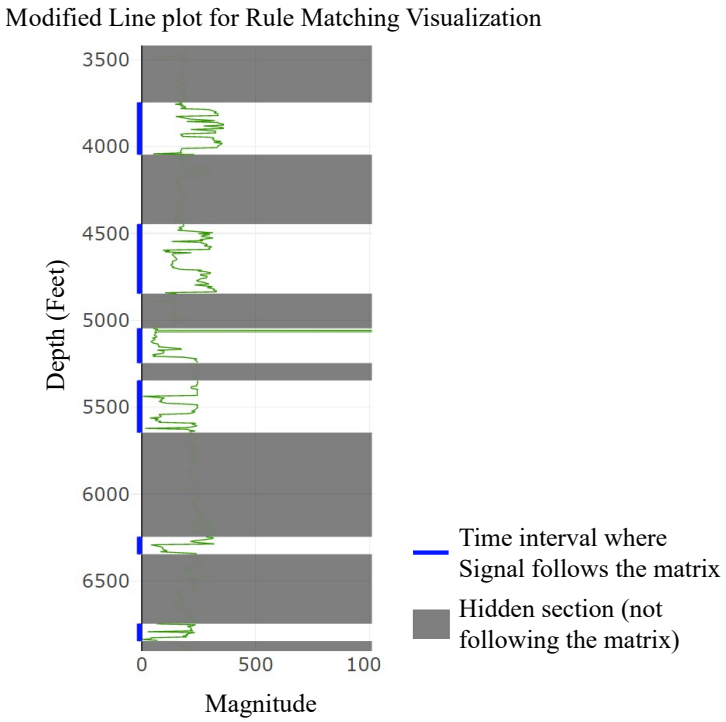

Figure 12: Visualization of rule matching, section of the signal where given standard deviation is larger than 50 is highlighted and other sections are hide with shading. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

4.4 Visualization and Detection of Bore-Hole Gauge with the Extended Probability Envelope

The conventional approach to casing wear visualization is to use 3D rendering and manually monitor the sections where wear occurs. This approach has limitations like obstruction and lack of details. Although 3D rendering highlights regions of over gauge with colors, the user needs to manually rotate and shift the 3D scene to find out the interesting sections.

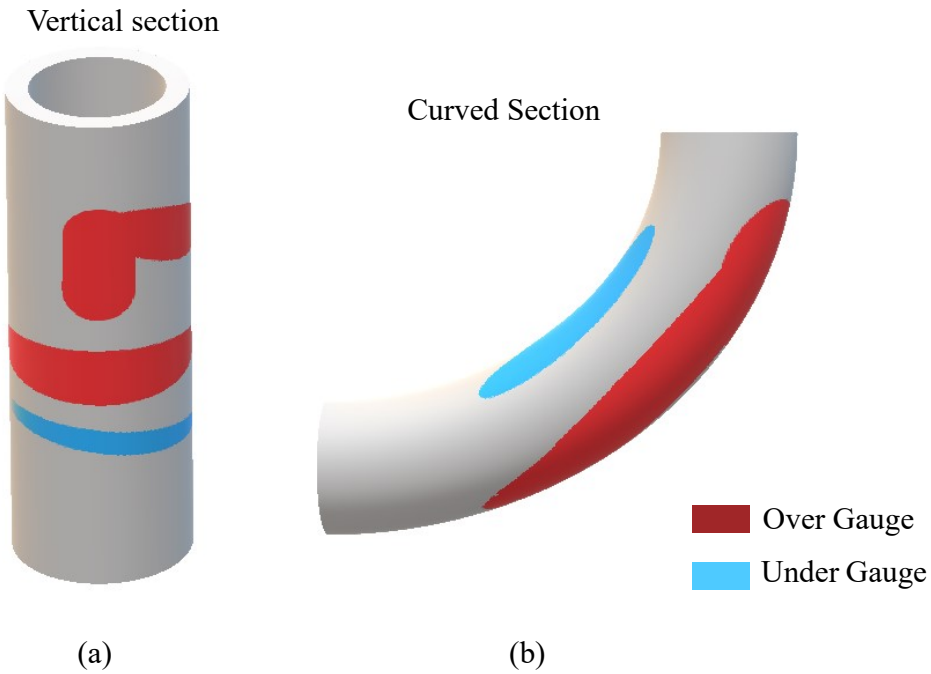

Figure 13: Illustration of 3-dimensional visualization of Bore Hole Gauge.

4.4.1 Naive Approach

The naive approach to visualize gauge (casing wear) with caliper log data is to use multiple connected area plots providing vertical casing view from different angles (directions). This is demonstrated in Figure 14. Radius measurements in opposite directions are plotted together as an area plot to create a vertical casing view (Figure 14 (b)). Each plot shows casing from a particular direction as described in Figure 15. Depth ranges within which over/under gauge exists can be easily captured with these plots along with their direction information.

For Caliper having many measuring fingers (typically 20-80 fingers), this approach will generate many plots. Manually analyzing and establishing correlation between such a large number of plots is labor intensive. To overcome this limitation, a customized approach with the help of PE-technique is proposed that can summarize a large number of radius measurements, while providing interesting details to focus at the same time.

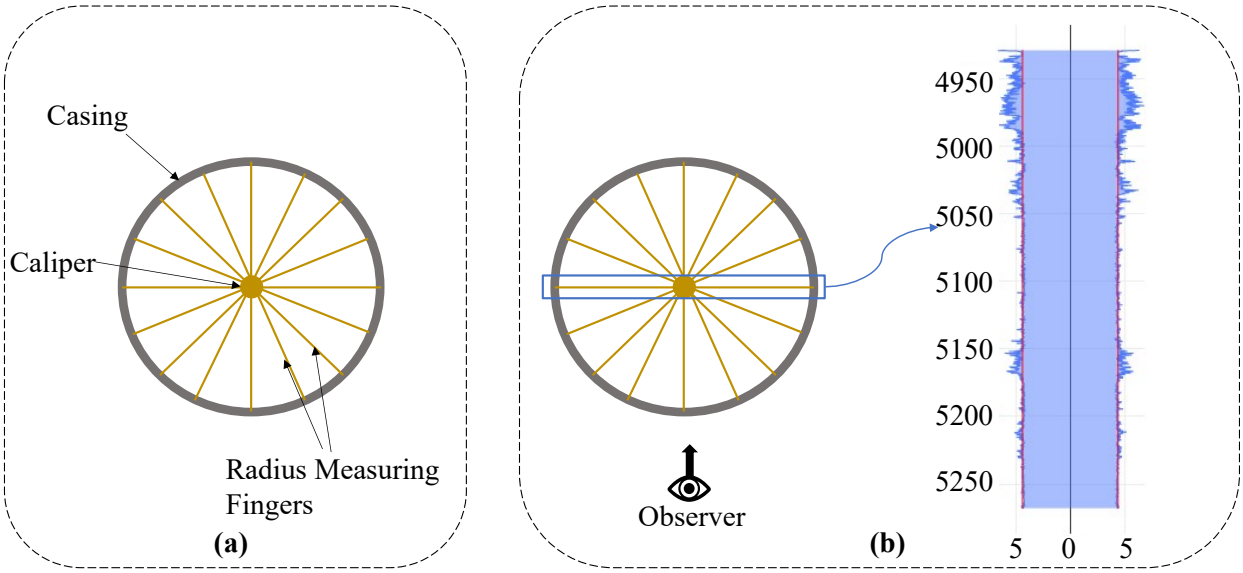

Figure 14: (a) Top-down view of casing and demonstration of Caliper Log data collection (b) Conversion of Caliper log data to vertical casing view (one plot per diameter)

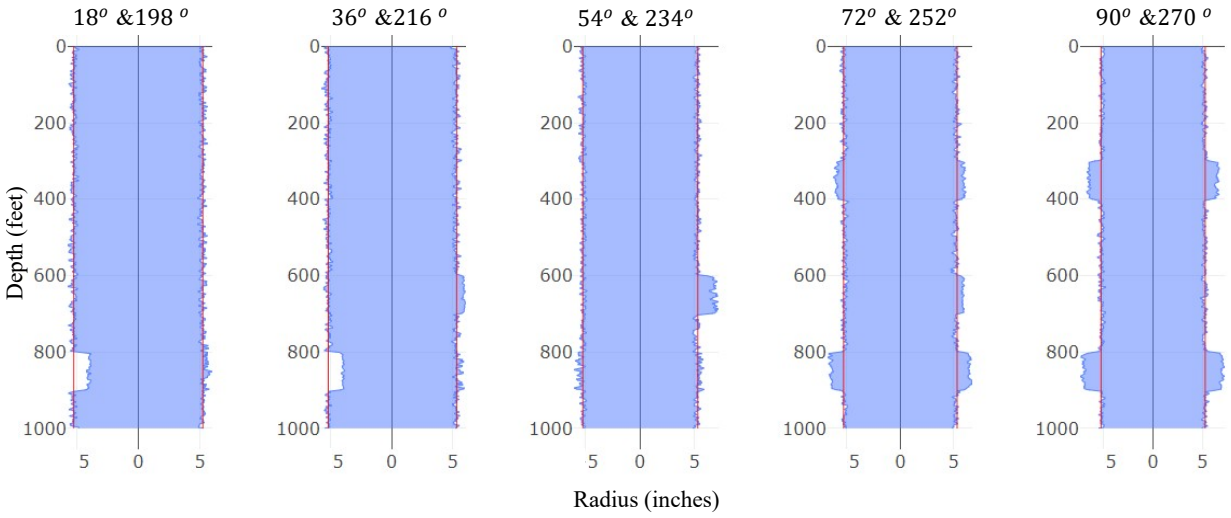

Figure 15: Vertical casing view from different directions, directions are represented in the form of rotation of the two radius from the reference in degrees.

4.4.2 Customized Approach Using the PE-technique

To overcome the limitations of the conventional and naive approaches, a customized approach with the help of PE-technique is developed which can provide summarized visualization of over/under gauge and focusing on parameter-specific interesting details. A user has controls to specify the parameters (e.g., the ideal radius of the well). Radius measurements for the entire well within a depth range are summarized using probabilistic envelope based visualization.

Envelopes are created based on probability (number of points) of measurements falling in certain region for a given depth range. It provides users an option to specify an ideal measurement for radius/diameter and a tolerance amount. Based on this information, it constructs an ideal region envelope (e.g., the envelope with purple color in Figure 17 (a)). It then constructs the envelopes for the radius measurements around the ideal envelope (e.g., the envelopes with different shades of the blue color in Figure 17: lighter color indicates fewer number of measurements falling in that region). This plot provides qualitative evaluation of the well for a specific depth range. At the center of envelopes an angle is provided to show the overall orientation of the well.

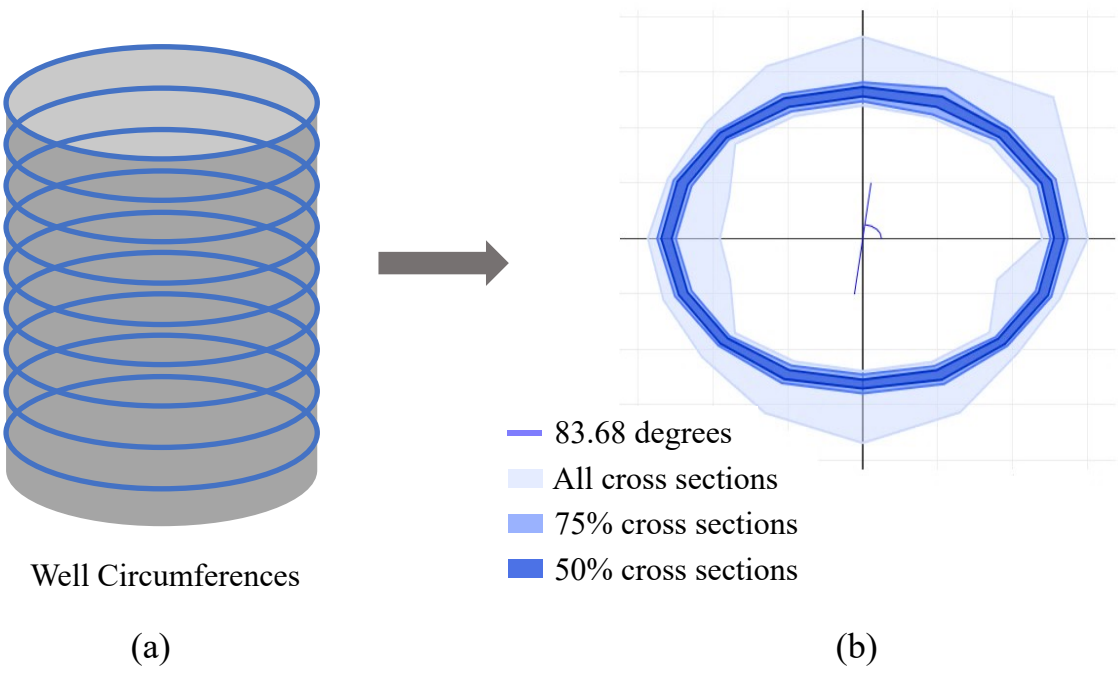

Figure 16: Summarized visualization technique using Probabilistic Envelopes for bore hole cross sections.

To provide the quantitative evaluation of the over/under gauge along with directional information, we also construct a radial bar chart connected to envelope plot, as shown in Figure 17. The bars outside the casing (positive bars) represents over gauge and bars inside casing (negative bars) represents under gauge. The lengths of the bars represent the amount of over/under gauge in terms of physical length measurement, and the color shows the amount of observations causing the over/under gauge in that direction. This is demonstrated in Figure 17 (b).

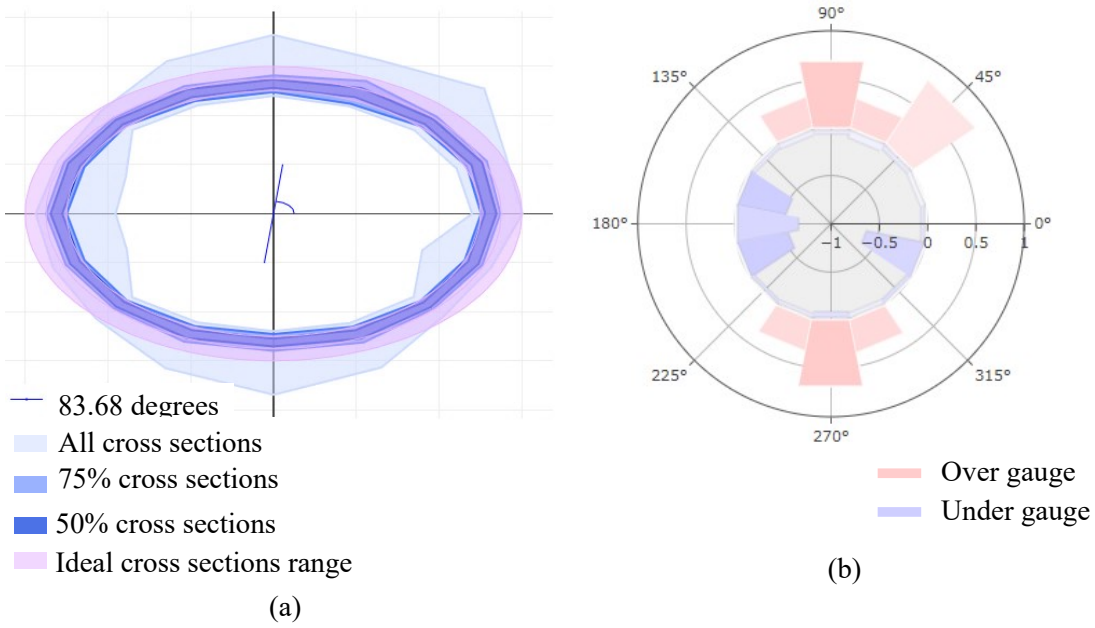

Figure 17: Over/Under gauge visualization with the help of envelopes.

Detection of Over/Under Gauge:

To detect over/under gauge, it compares the circumference at each measurement with the ideal circumference$\pm$*tolerance* specified by the user. The circumference is plotted as a vertical discrete heat-map where the red color region shows over gauge, the blue color region shows under gauge and in gauge is represented with the white color as shown in Figure 18 (b).

Margin - Let M be the margin value, it marks a depth range as out of the gauge only if the measurements in heat-map are out of the gauge continuously for M feet. With this margin (or threshold) value, small over/under gauge ranges that may be caused my measurement errors can be filtered. It does not mark individual measurements as over/under gauge, but it looks for depth ranges larger than M feet which has all the measurements falling out of the gauge.

Users have an option to select the median circumference as the ideal circumference instead of specifying one explicitly. Users can also modify the margin. Figure 18 explains how the heat-map

is obtained. To highlight the section of depths where over/under gauge occurs, it shows its outlines with the red color.

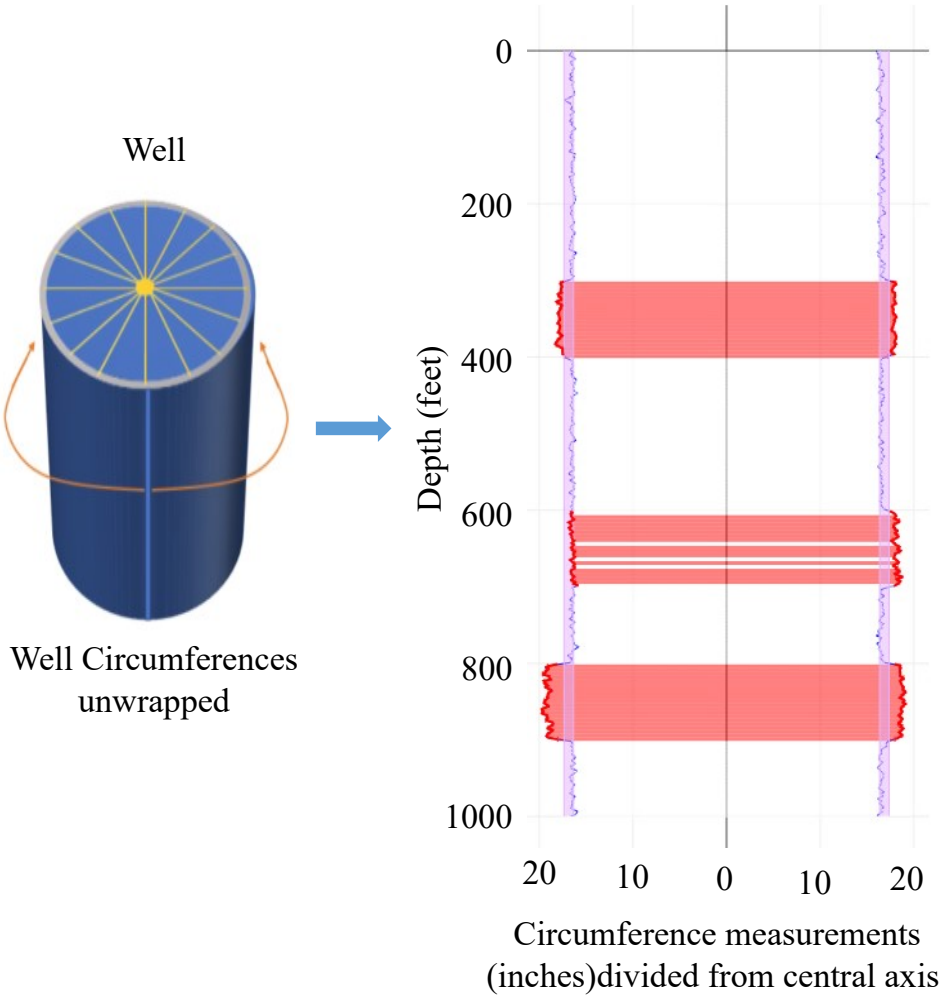

Figure 18: The circumference is computed based on the radius measurements and is unwound to create a vertical heat-map. The magnitude of circumference is compared with the ideal circumference and over gauge segments are marked with red color. Under gauge are marked with blue color (if any).

Once the heat-map is obtained and the depth ranges where over/under gauge occurs are detected, we use the above envelope visualization to plot the statistics of the circumvents within each

of these depth ranges to represent the amount and orientation of the over/under gauge in that depth range. Figure 19 demonstrates this. Specifically, a depth range with over/under gauge is selected, all the radius measurements within this range are then used to create an envelope visualization (Figure 19) to provide the detailed visualization of the over/under gauge within this depth range. Based on this visualization shown with red color, the user can intuitively identify the amount of over/under gauge (based on the distance between the red circle and the purple/ideal circle) and the orientation of the bore hole gauge (i.e., in the directions of 91.58°, 45.09°, and 87.4° in this example). With this functionality, our tool can automatically detect the sections where over/under gauge occur and plot them individually for detailed inspection.

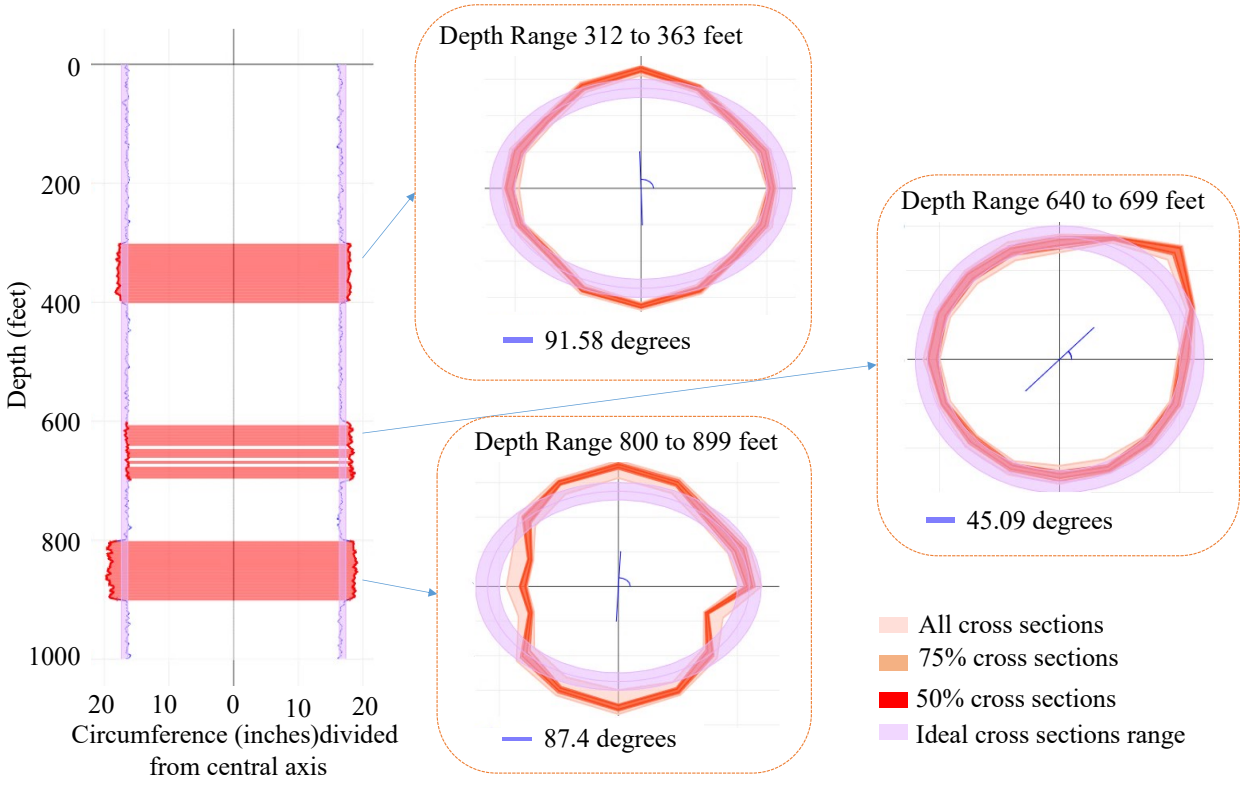

Figure 19: Detection of depth ranges of Over/Under gauges and their detailed evaluation. Detected depth Ranges with over/under gauge from the heatmap, are all individually plotted with envelope plot for detailed evaluation.

4.4.3 Evaluation of the Customized Approach

The probabilistic envelopes are slightly modified and made circular to be used in this approach. The time complexity of the computation of these envelopes is similar to that discussed in sec 4.1.2. Here, all the radius measurements at a depth level can be considered similar to a reference. Consider n (number of fingers in measuring caliper) radius measurements for each observation, depth of bore hole as D and the depth-step for which observations are repeated is d units then the run-time complexity $O(A)$ is,

$$O(A) = O\left(n \times \frac{D}{d}\right)$$

As explained in sec 4.1.2, time-complexity does not depend on the number of layers or probability values assigned to each layer.

Again, there is a trade-off between speed and accuracy because the run-time complexity is proportional to the number of radius measurements at each observation while having large number of radius measurements at each observation also enhances the visualization and accuracy of well structure.

fig:Eval$_B$ $HGshowsprobabilisticenvelopeplotsforthedatahavingradiusmeasurementfor8and16directionsatea$

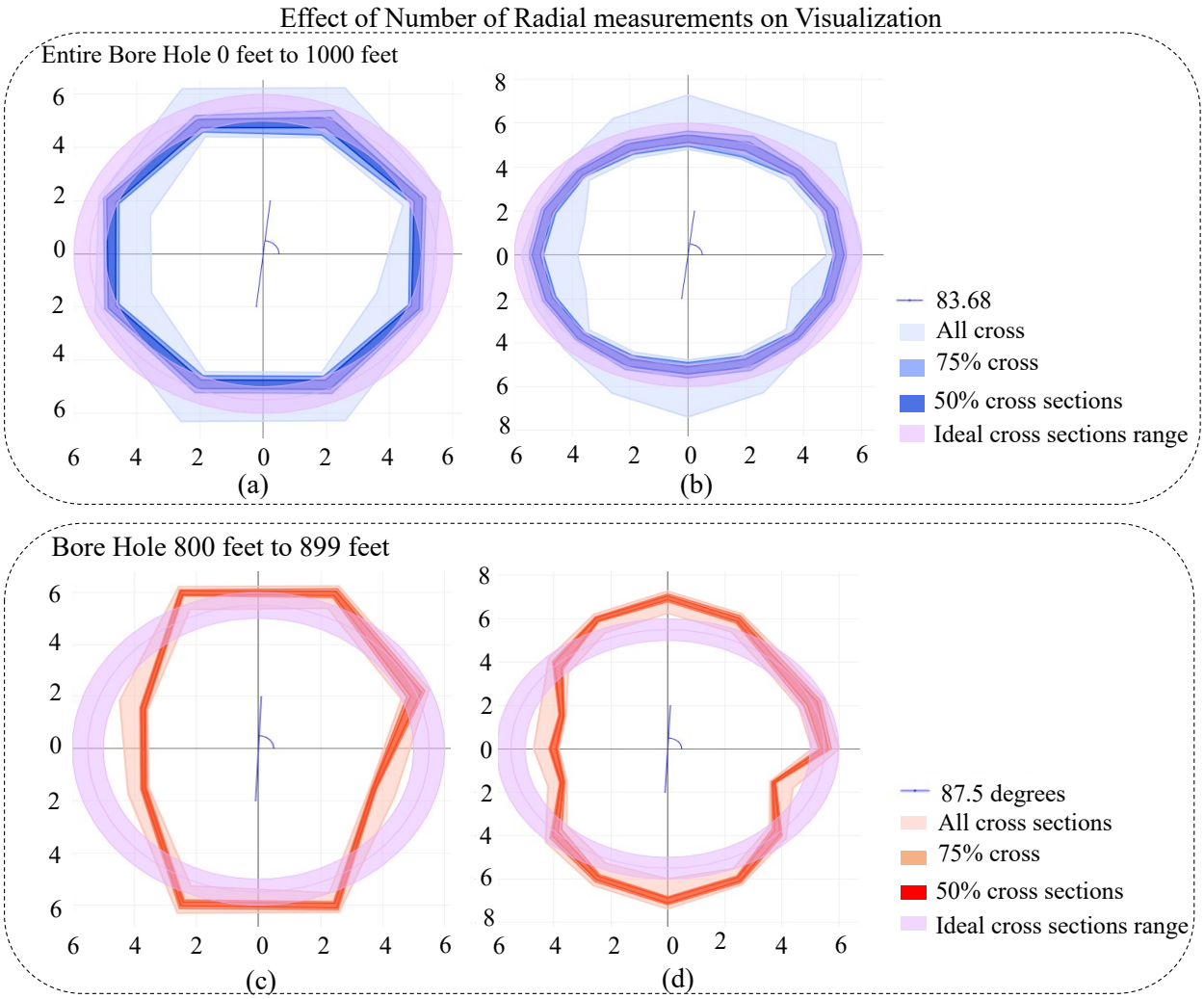

Figure 20: Bore Hole gauge visualization with the help of circular probabilistic envelopes. (a) and (b) show the entire bore hole, while (c) and (d) show the detected depth range 800 to 899 feet depth where sever wear occurred. (a) and (c) have the radius measurements in 8 directions whereas (b) and (d) has radius measurements in 16 directions with uniform angle interval, at each depth level.

5 Synthetic Framework for Time Series Data

There are many possible variations between two time-series data (describing two signals) processed by a matching algorithm. It is difficult to have all such different variations in a single pair of real-life time series data and knowing the variations between them may help us better understand how these variations impact the performance of different DTW measures in different tasks. This leads us to create a synthesis framework capable to generate pair of time-series data with desired features and variations. In this section, I will describe how the synthesis framework generates realistic time-series and series pairs with controllable variation for the subsequent evaluation. I also assess the effectiveness of our synthesis framework by using it to fit a number of real-world signals.

5.1 Realistic 1D Signal Generation Technique

A linear temporal signal can have different features, but here mainly three of them are considered, that is, the range of its magnitude, the distance between two successive peaks/valleys and the nature of function it follows between peaks/valleys. Considering these three main features, a synthesis framework which is capable to create realistic time-series is developed.

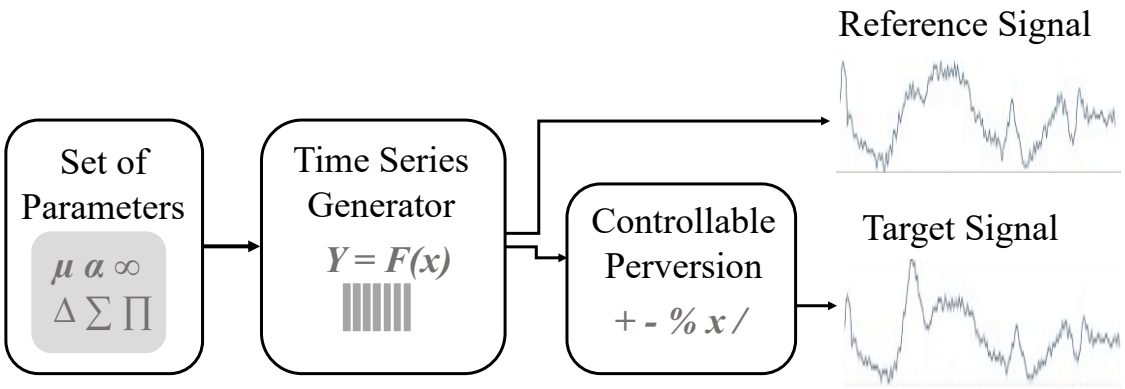

Figure 21: Our time series synthesis framework consists of two steps. The first step produces an initial series, while the second step deforms the initial series with the controllable variations to generate the second series.

Input parameters: This synthesis framework provides users the control over the range of magnitude of the signals (Low, High), the allowed range of distance between two successive peaks/valleys (p_1, p_2), length of the signal (L) and the list of allowed functions which can be used to interpolate the signal between successive peaks/valleys. In the experiments shown in this thesis functions of the form $y = cx^a + dx^b$ are used where a and b vary between 0.1 and 2 randomly and (c,d) are coefficients fitted according to the location of end points (peaks/valleys).

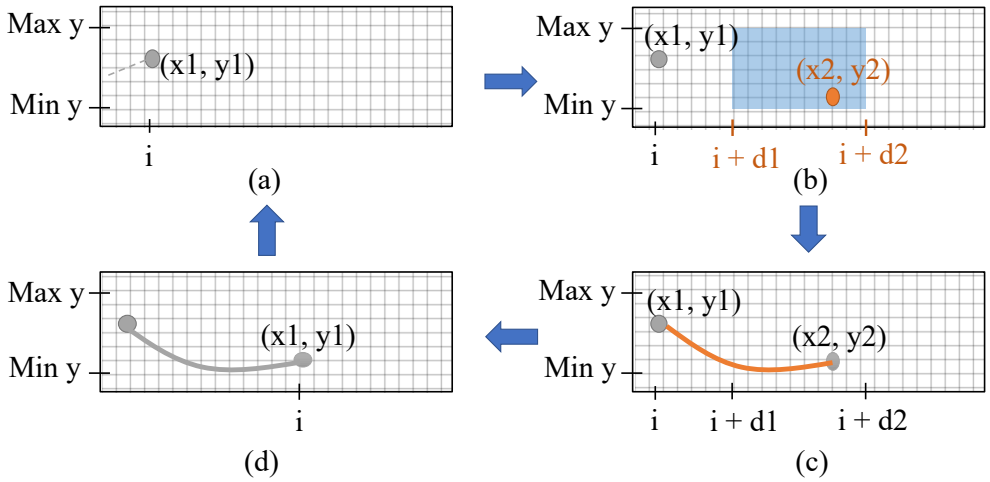

Figure 22: Time Series Generator (a) Sample a point between minimum and maximum magnitude (b) Randomly sample another point between minimum and maximum magnitude and place it at random distance between maximum and minimum allowed distance between two peaks/valleys from first point (c) Generate a random function like $y = c_1x^a + c_2x^b$ and fit it between these two points (d) Add these sampled points to the signal array and repeat this process until the desired length is reached.

Generation: After getting these specifications from the user, our framework generates a signal which may not look realistic initially (fig:genInitial(a)) but then it adds random noise with $+-20\%$ of the magnitude range (High-Low) (fig:genInitial(b)). In particular, our framework creates a point (X_1, Y_1) in the range of (Low, High), it then creates a rectangular region spanning the range of (Low, High) and time axis range (p_1, p_2) as shown with blue shade in fig:generateinitialseries (b) to

sample another point (X_2, Y_2). It then randomly selects a function from the user specified functions to fit the signal between (X_1, Y_1) and (X_2, Y_2). This process is repeated until the desired length is reached. This process is described in Algorithm 2 and fig:generateinitialseries.

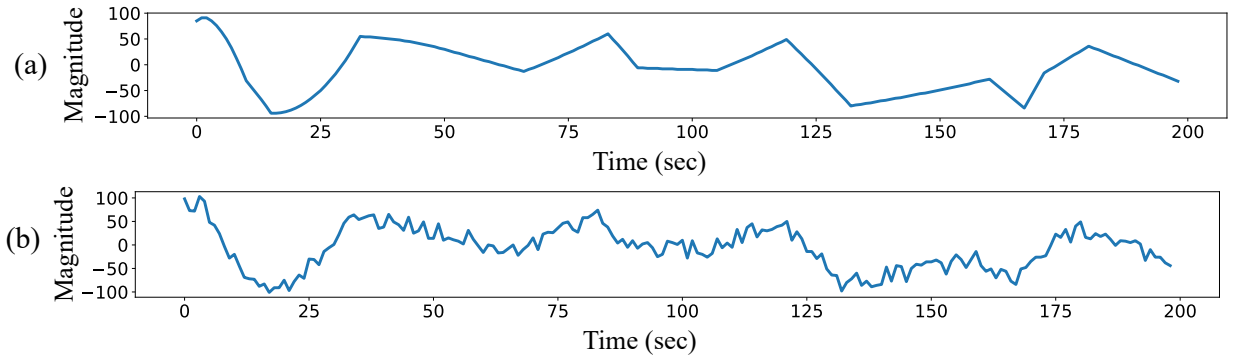

Figure 23: Initial synthetic signal generation without (a) and with noise (b) inserted. Note that synthetically generated data does not have a physical unit for magnitude.

Algorithm 2 Algorithm *GenerateSignal*.

Input: Length of Signal (L), Range of Magnitude for Sampling (Low, High), Min and Max distance between two peaks/valleys (P_1, P_2).

Output: Synthetic Signal.

Initialize an empty list S
PrevY = pick a random integer between Low and High
PrevX = end = 1
Append PreY to S
while $end < L$ **do**
    $RangeStart = PrevX + P_1$
    $RangeEnd = PrevX + P_2$
    if $RangeStart \geq L$ **then**
        Stop and return S
    if $RangeEnd \geq L$ **then**
        $RangeEnd = L - 1$
    x = pick an integer between RangeStart and RangeEnd randomly
    y = pick an integer between Low and High randomly
    Pick a two numbers a and b between 0.1 and 3 and create functions as $y = c_1x^a + c_2x^b$ and $PrevY = c_1PrevX^a + c_2PrevX^b$
    Solve the above equations for c_1 and c_2
    Fit the function $y = c_1x^a + c_2x^b$ between PrevX and x and append the y values to S
    PrevX = x
    PrevY = y
Initialize empty list N
for i in range 0 to Len(S) **do**
    $t = randomInt(0, 0.4 * High) - 0.2 * High$
    append t to N
S = S + N
Return S

5.2 Controllable Distortion to Reference Signal

Next, the framework uses the initial signal generated above as a reference and produces another signal (target) by perverting (or modifying) this reference signal. The overall broad pipeline of this framework is shown in fig:SystemArchitecture. There are two basic deformations/features which can be composed to create other variations. Using these basics and composite variations any temporal signal can be transformed to any other temporal signal; this will be detailed in later sections.

5.2.1 Scaling the Signal

The first basic modification considered here is the scaling. This scaling will modify the length of a portion of the reference uniformly. This scaling may change the length of the target (or output) signal compared to the reference. It provides an option for the user to decide whether they want to ensure the identical length between the target and reference signals or not.

Input parameters: The scaling takes the beginning of the window for scaling, W_0, and its ending, $W1$, as well as a scaling factor s (usually ranging from 0.5 to 1.5) as the input.

Scale a Random Portion of the Signal Let the target signal created after the controlled deformation (scaling) be y and the reference signal be x, when the window from index W_0 to index W_1 is scaled with the scaling factor of s. Then the magnitude of the signal at j^th index of y is taken from the i^th index of x where i is determined using the following formula.

$$i = \begin{cases} j, & \text{if } j < W_0 \ . \\ W_0 + (j - W_0)s, & \text{if } W_0 \leq j \leq W_0 + (W_1 - W_0)s. \\ (W_1 - W_0)(1 - s) + j, & \text{otherwise.} \end{cases} \tag{1}$$

Here, the window from index W_0 to W_1 on the reference signal is scaled by the scaling factor (s), and the later section of the sequence is shifted to accommodate this scaling. fig:Scaling shows this variation between the sequences. Gray boxes show the scaling window in both the reference

and target signals. In this particular instance the window is scaled up by 137%.

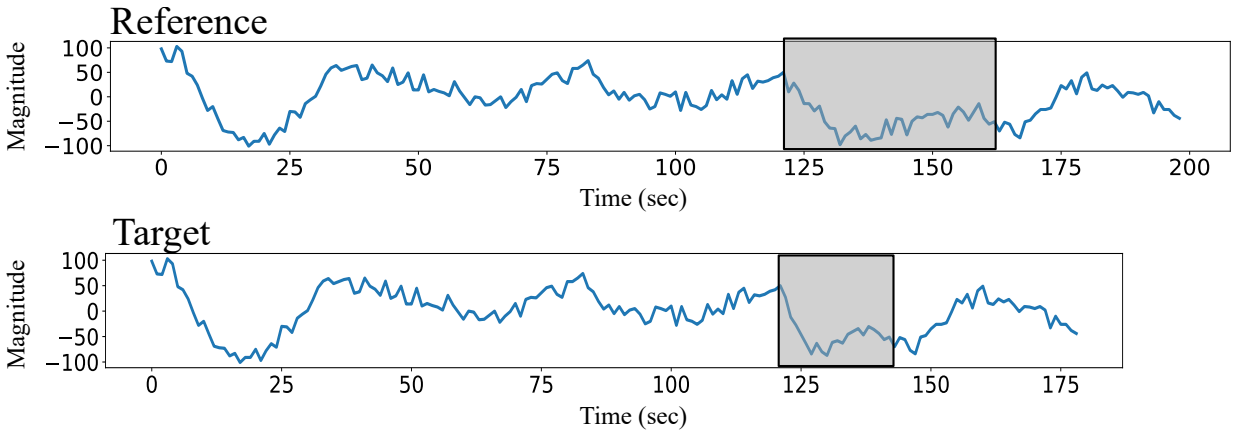

Figure 24: Scaling the shaded region without length preservation. Note that since the plot represents synthetic data, magnitude does not have a physical unit.

Scale a Random Portion while Preserving Signal Length Time series data sequences often have their features shifted even if they represent similar physical quantity and have the same length. That said, if a portion of the reference is scaled, other portions also need to be scaled and shifted to preserve the length. To achieve that, let the target signal created after controlled deformation on reference signal x be y. When the window from index W_0 to W_1 is scaled with the scaling factor of s and remaining sections with s' to keep the size of target signal same as the reference. Let the length of original signal be L, and length of scaling window (W_0, W_1) be l then s' can be computed as $s' = \frac{L-l\times s}{L-l}$. Then the magnitude of series at j^{th} index of y is taken from the i^{th} (where i is rounded to the nearest integer) index of x where i is determined as follows.

$$i = \begin{cases} js', & \text{if } j \leq W_0 s' \ . \\ \frac{W_0 s'(s-1)+j}{s}, & \text{if } W_0 s' < j < W_0 s' + (W_1 - W_0)s. \\ \frac{(W_0 s' + (W_1 - W_0)s)(s'-1)+j}{s'}, & \text{otherwise.} \end{cases} \tag{2}$$

fig:FullScaling shows such an example of scaling with the preserved length. The gray boxes

highlight the portion of the signals that was adjusted after the initial scaling shown in fig:Scaling.

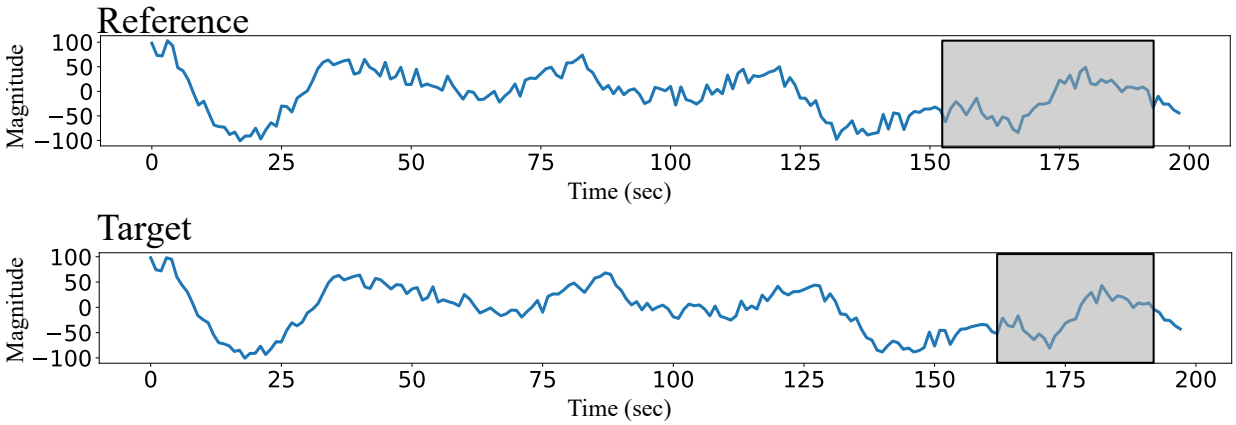

Figure 25: Length preserving scaling. Adjustment is made in the shaded region. Note that synthetically generated data does not have a physical unit for magnitude.

5.2.2 Addition of Random Gaussian Peaks to the Time Series

Sometimes, two time series sequences have similar features at similar locations on the time axis, but they may have some extra features or features are missing on one of them. This variation is sometimes due to noise or human errors. Simply increasing or decreasing the magnitude at a single point to achieve the above variation is usually not sufficient, as it may be equivalent to inserting a random noise, making the resulted signal unrealistic. To address that, our framework adds a Gaussian peak, i.e., it increases the magnitude of the signal within a small window with its center having the highest magnitude and gradually decreasing away from the center. fig:RGP provides such an example.

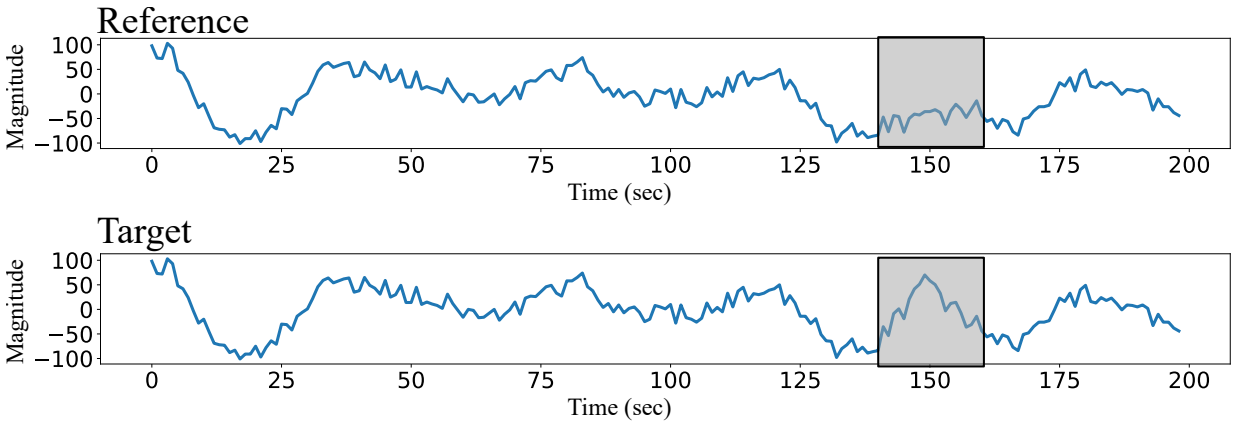

Figure 26: Adding a random Gaussian peak in the shaded portion. Note that synthetically generated data does not have a physical unit for magnitude.

Adding a random Gaussian peak **(RGP)** to the signal causes changes in local features of the signal; this type of peaks is sometimes due to noise or human errors. Simply increasing the magnitude at a single point is usually not sufficient, as it may be equivalent to inserting a random noise, making the resulted signal unrealistic. To address that, our framework adds a Gaussian peak, i.e., it increases the magnitude of the signal within a small window with its center having the highest magnitude and gradually decreasing away from the center. Instead of just one peak, multiple random Gaussian peaks **(MRGP)** can also be added at random places on the signal. This will create a signal with more variation. Let us say our Gaussian peaks are $\{g_1, g_2, g_3 ... g_n\}$ and we add this to the signal y at index i, then the index window where this peak will be added is (W_0, W_1) $W_0 = i - \frac{n}{2}$ and $W_1 = i + \frac{n}{2}$.

$$y_j = \begin{cases} x_j, & \text{if } j < W_0 \text{ or } j > W_1 \ . \\ x_j + g_{(j-W_0)} & \text{otherwise.} \end{cases} \tag{3}$$

5.2.3 Scaling along with Addition of Gaussian Peaks

This type of variation is widely observed in real world time series data. Usually, two similar time series data sequences measuring similar events (e.g., drilling well signals) have shifting of features along with some additional features or some of the features are missing on one of them. This type of variation in sequence can be considered as the combination of the previously described two variations (i.e., shifting and addition/subtraction of peaks).

To achieve the combination of shifting and RGP insertion or removal, first the reference signal is scaled similar to **modification 1**, then a Gaussian peak is added at random place on this scaled signal similar to **modification 2**. This new perverted signal has much more variations compared to those produced by the previous modifications, since it changes both temporal property by scaling and also local features by addition of peak(s). The mathematical formula is similar to the one described in sec:modification2, but this time instead of adding it to the reference, the peak is added to the scaled signal.

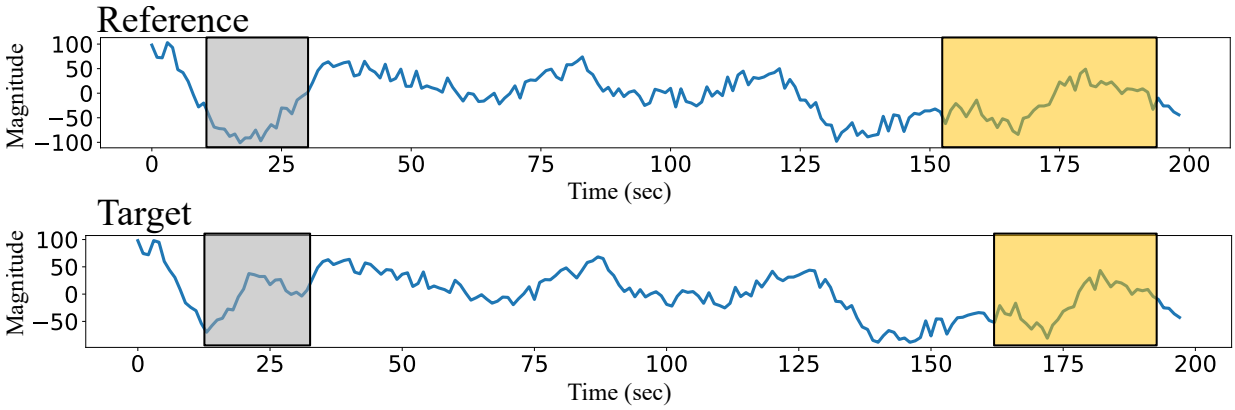

Figure 27: Modification by both scaling (in the yellow shaded region) and adding a random Gaussian peak (in the gray shaded region). Note that synthetically generated data does not have a physical unit for magnitude.

5.3 Effectiveness of the Synthesis Framework

To prove that using only the previously described modifications, any temporal signal can be transformed to another temporal signal, a simulated annealing [11] based framework is developed to optimize the parameters used by our synthesis framework to enable the generation of a synthetic signal that is sufficiently close to the real one.

Take two signals as input. Without loss of generality, it considers one as the source and the other as the target. The goal is to transforms the source signal to make it look like target as close as possible by performing only the above described modifications. It uses Euclidean distance as a measure to evaluate how close two temporal signals are. Basically, simulated annealing uses Euclidean distance between the source and generated target with the current parameters as optimization function and tries to minimize this function.

There are two steps in this our fitting framework. First, scaling is performed on the source signal using simulated annealing. It is mainly used to resize the source in order to match the length of the target signal. There are three parameters to fit in this process, location of scaling, width of scaling window and scaling factor (magnitude of scaling). To simplify and speed up this search, the width of scaling window is set to 50% the original length of source. Let the width is W then, $W = 0.5 \times length(source)$. Another parameter scaling factor is computed using below formula,

$$s = 1 + \frac{length(target) - length(source)}{W}$$

The third parameter, i.e., location of this window is found by via simulated annealing by minimizing the Euclidean distance between source and target. Once source is scaled and its length matches the length of target, it is passed to the second step.

Figure 28: Signal-fitting result (synthetic data does not have a physical unit for magnitude).

In the second step, multiple Gaussian peaks are added on source with the help of simulated annealing. Simulated annealing is performed multiple times to reduce the Euclidean distance between the two signals, and in each iteration one peak is added. There are also three parameters to optimize, including the location of the peak, width of the peak, and magnitude of the center of the peak. All three parameters are initialized randomly and optimized by simulated annealing for each Gaussian peak. Multiple peaks are required to be added in order to match the curves on the target. It keeps adding the peaks until the Euclidean distance between source and target is smaller

than threshold (T). It computes a threshold which is proportional to length and magnitude range of the target signal as shown below.

$$T = \frac{x}{100} \times magnitude(target) \times length(target)$$

Here x can be used to control the accuracy of fitting. I performed experiments to evaluate the above signal-fitting framework. In these experiments, I fixed, $x = 1$, i.e., 1% error is allowed. fig:SA$_{R}esult1showshowthefittingprocessusingtheproposedsynthesisframeworkcandeformthesourcesignaland$

There is a trade-off between the number of peaks added/subtracted and the error between source and target. In a noisy signal pair, while transforming source to look like target, few initial peaks after scaling transforms overall trend of source to look like target. Once large peaks are adjusted, the framework starts adjusting small peaks and at some point, after two signals almost have identical trend, it tries to adjust individual (or two adjacent) point/s to make source look exactly like target. Allowing a large number of peaks to be added/subtracted by the framework, it is capable to completely transform a source linear signal to exactly look like target linear signal. But in general, we do not wish to use a large number of peaks to deform the source to target. In our experiment, we found the number of the peaks needed to produce a signal with 1% distance from the target is at most half of the number of samples on the source.

6 Evaluation of Varying DTW Matrix

6.1 DTWs for Signal Alignment

With the synthetic signal pairs with the known variation between them, we now perform a number of evaluations on different DTW measures. In particular, we produce sets of synthetic signal pairs with one type of variation (e.g., scaling only, one RGP, combined scaling and RGPs, etc.) and perform signal alignment between the individual pairs using different DTW measures. We quantitatively measure the alignment accuracy given the known correspondence between the samples on the corresponding pairs and report the performance of those DTW measures.

As mentioned earlier, aggregate distance over magnitude (ADM) (i.e., the Euclidean distance) between two signals is usually applied to measure the accuracy of the alignment. In our experiment, since the ground truth alignment is known, the aggregate distance over time (ADT) that aggregates difference of time value of matching points is proposed. Both ADT and ADM measures are used for performance evaluation shown in Table-1. In the following, the discussion is organized based on the type of variations that is imposed to the signal pairs. For each group of experiments, 50 signal pairs with varying parameter values were generated and utilized, and a representative result is selected for each group for the discussion.

6.1.1 Scaled Signal

When signal features are shifted on time axis with no variation in magnitude, standard DTW methods (DTW, and DDTW) seems to be working best among these four variants. In fig:Result$_S caling(a) when weight parameter(g) is not optimized weighted methods usually per forms worse th$
$29(b) weight parameter(g) is optimized, both WDTW and WDDTW per form almost similar to DTW and DDTW as in$

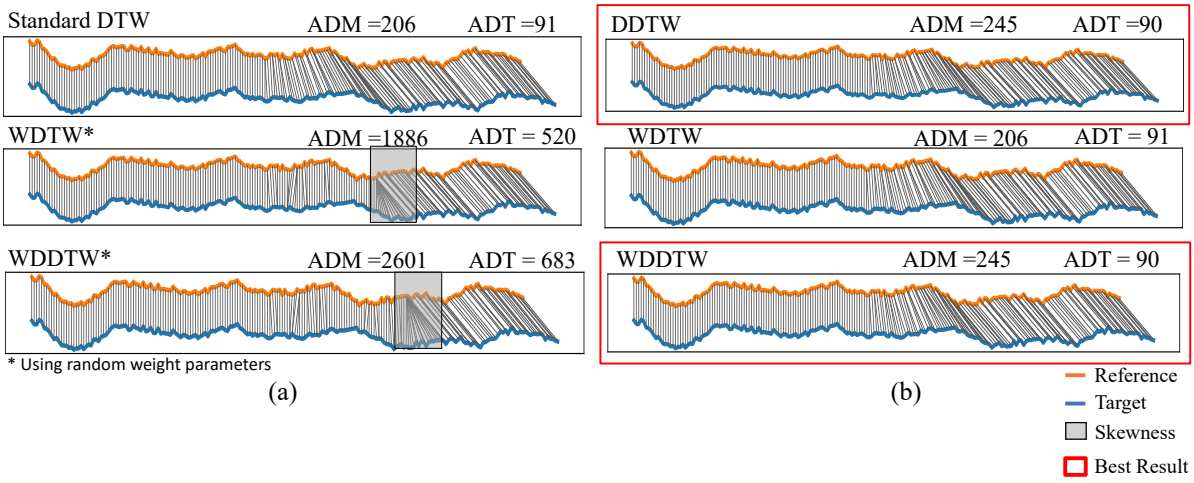

Figure 29: Results from different variants of DTW on signal pairs where target is simply scaled on time axis at certain places. (a) Weight parameter [19] (g = 0.4, 0.4) of WDTW and WDDTW are simply selected at random. (b) Weights parameter (g=0.21, 0.11) are optimized using Monte Carlo sampling method for both WDTW and WDDTW. Note that we do not show axis labels as the visualization shows the alignment information.

In the remainder of the results shown in this work, optimized weight parameter (g) is utilized for all different signal pair variants.

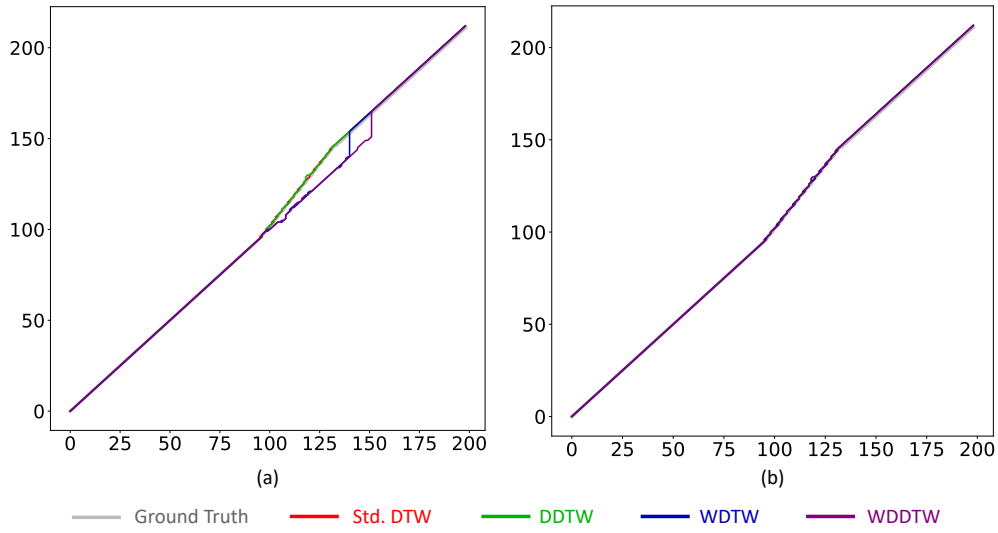

Figure 30: Warping Path (a) With randomly selected weight parameter (g) both WDTW and WDDTW deviates from ground truth at the central region while both DTW and DDTW follows almost accurate path with some staircase effect to accommodate scaling. (b) With optimized weight parameter (g) WDTW and WDDTW along with standard methods follows almost accurate path. x and y axes represent indices of the reference and target signals, respectively.

6.1.2 Scaled while Preserving Signal Length

Next, the target signal that has only shifted features on time axis with no or negligible variation in magnitude and length compared to the reference signal is considered, as shown in fig:FullScaling. There is not much difference in ranking the DTW methods for the alignment with both the ADT and ADM distance measures in this type of series variation. On ADT distance measure, DDTW seems to outperform others with a small margin. As in fig:Result$_FullScaling, the alignment results from these DTW variants$

For aligning the time series pairs with this type of variation, DTW tries to warp the scaling over time axis by matching points at different index on two sequences thus minimizes the total difference between magnitude values of matching points. Weighted DTW, with their weight parameter optimized can also accomplish the same level of accuracy but they could not outperform standard DTW and DDTW.

For the alignment problems when we know there's shifting of features over time axis and not much difference in the magnitude of the sequences, the recommendation is to use standard DTW methods (either DTW or DDTW) and not weighted variants, because they may perform worse with random weight parameters and optimization of weight parameter is itself a very costly process. Even after optimizing the weights, they are not guaranteed to perform better than the standard DTW methods.

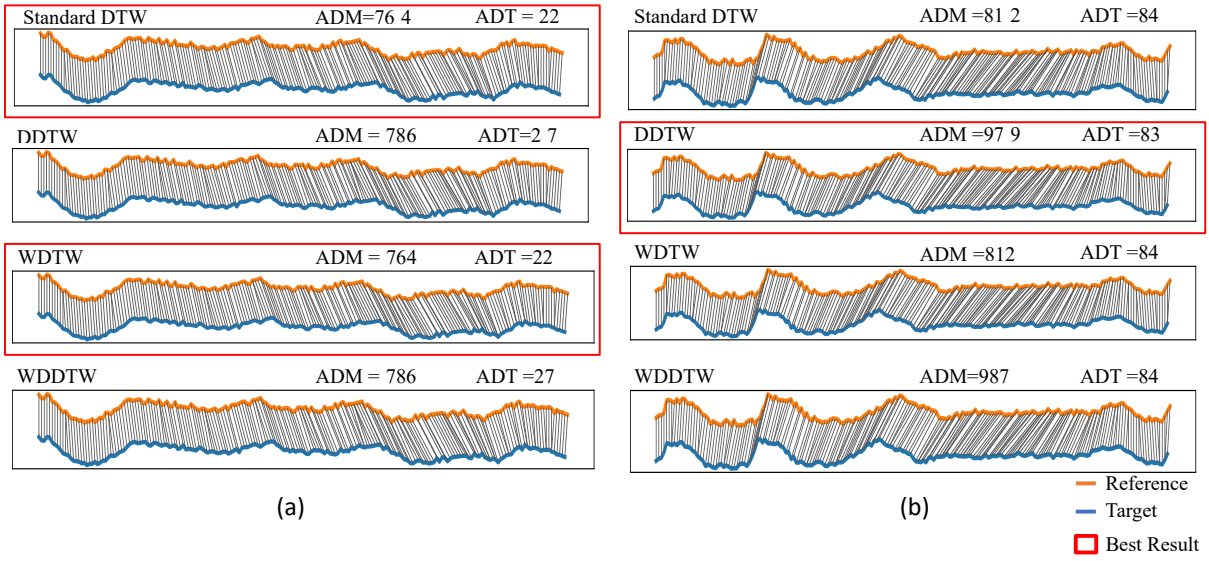

Figure 31: Alignment result from DTW variants when target time series has features shifted on time axis with negligible difference in magnitude than the reference. (a) and (b) shows two different instances. DDTW seems to be working best over ADT measure while Standard DTW performs best over ADM measure. WDTW(g=0.01) and WDDTW(g=0.02) with optimized weight parameter g are very close to DTW and DDTW but could not outperform them. Note that axis labels are not shown here, as the visualization aims to show the alignment information.

6.1.3 Random Gaussian Peak Added

When two signals have few differences in the features they contain, for example an extra peak or a missing peak, the widely used distance measure ADM fails. Since it tries to minimize the aggregate difference between the magnitude of matched points, it forcibly matches an extra peak with some

high magnitude point on the other signal. The different variants of the DTW are also evaluated using this distance measure, since it is widely used in the literature as well as it seems to be the best measure when we do not know the accurate alignment of two sequences. But distance measure in terms of ADT is also provided to evaluate the results given the ground truth alignment. As in fig:Result$_RGP(a)whentargethasoneextrapeakthanreference, weightedDTW(withoptimizedweightparameters)$ $weightedvariants(onADTmeasure), thoughthisoptimizationcomesatveryhightimecomplexity.Theycouldnotoutp$

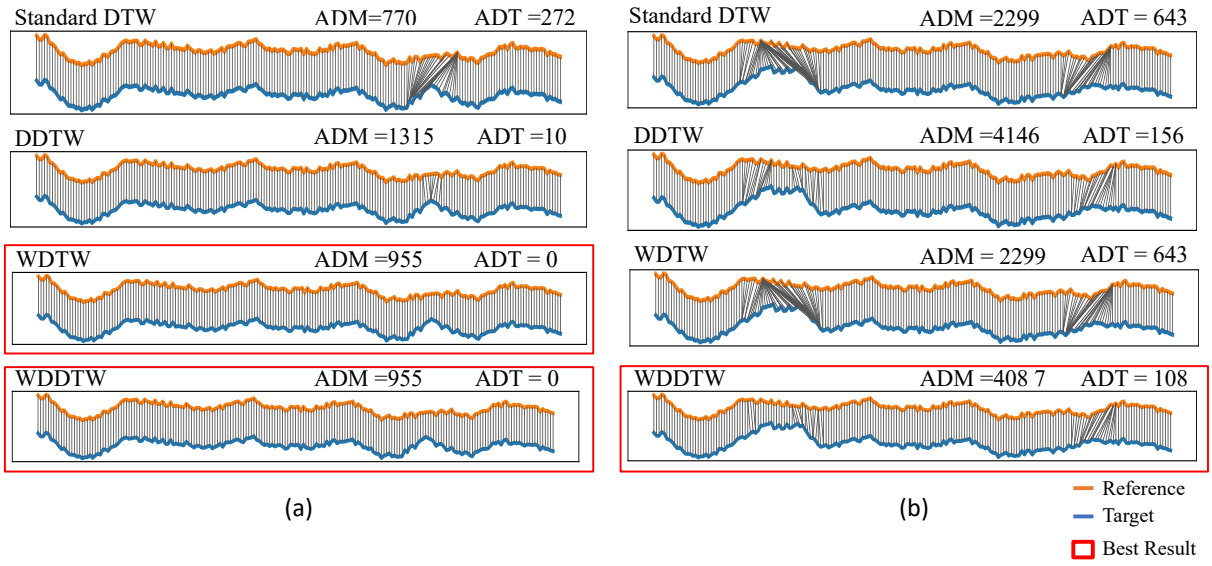

Figure 32: Alignment result from DTW variants when target has a random Gaussian peak added. (a) Target has only one random Gaussian peak (RGP), on the ADM measure Standard DTW performs best while on ADT measure Weighted DTW variant(s) with optimal weight parameter outperforms DTW and DDTW. (b) Target has multiple RGP, Derivative variants (DDTW and WDDTW) performs well on ADT measure while Standard DTW methods works best on ADM measure. Note that axis labels are not shown here, as the visualization aims to show the alignment information.

fig:Result$_RGP_Matrix(a)and(b)showthewarpingpathforthealignmentresultsshowninFigure-$ $32(a)and(b), respectively, whichbetterdemonstratetheADTmeasure.Withanextrapeakthatdoesnotaddontotheexi$ $32(b), thewarpingpathsforstandardandWDTWaresimilartothepathsthatDDTWandWDDTWfollow.Theyallf$

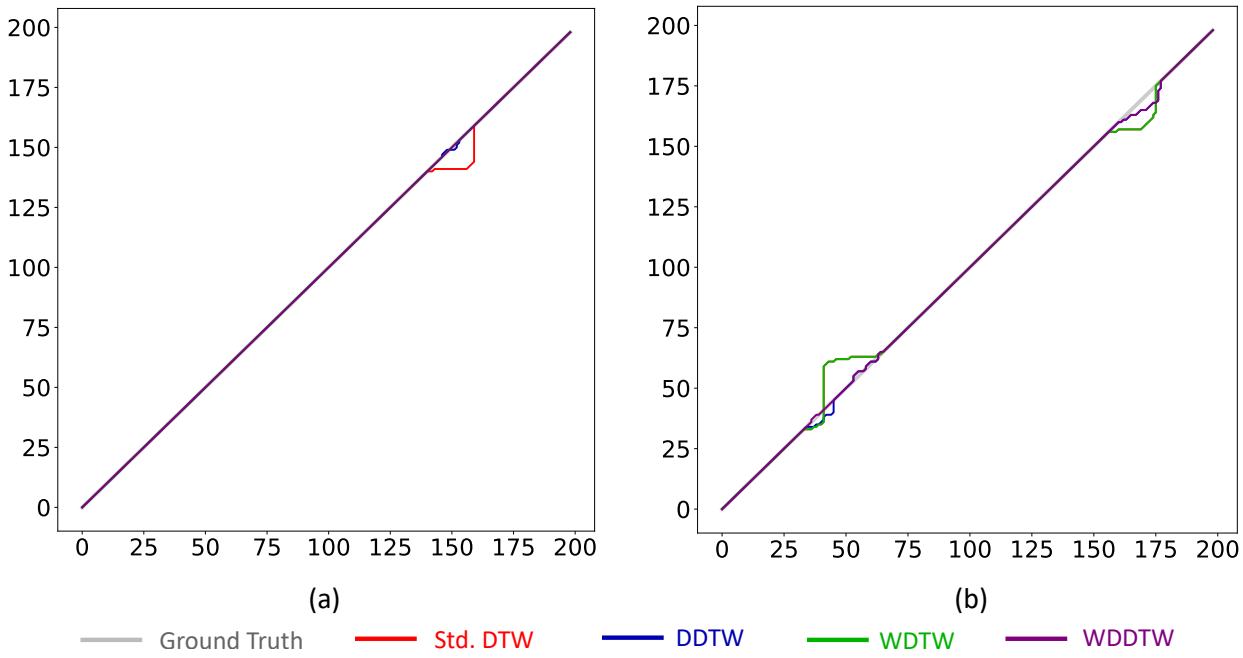

Figure 33: Warping path for the instances of time series alignment in Fig-32 (a) and (b) respectively. (a) With single RGP non-weighted DTW variants jiggles a little around the peak. (b) With multiple peaks weighted DTW could not accommodate all the peaks adjustments since weights are optimized on ADM measure, they try to minimize the magnitude difference and end up with those deviations from the true path. x and y axes represent indices of the reference and target signals, respectively.

6.1.4 Random Gaussian Peak Added on Scaled Signal

With this type of variation which is mostly found in real-world time series data, derivative DTW seems to be winning over all others. As shown in fig:Result$_S RGP(a) and (b), when the target has one or multiple peaks c$

This variation is widely seen in many real-world applications, such as formation transitions alignments, and streamline classification. **Based on our experimental evaluations, we suggest the usage of WDDTW for the alignment when the signal pair has both scaling and random peaks difference.**

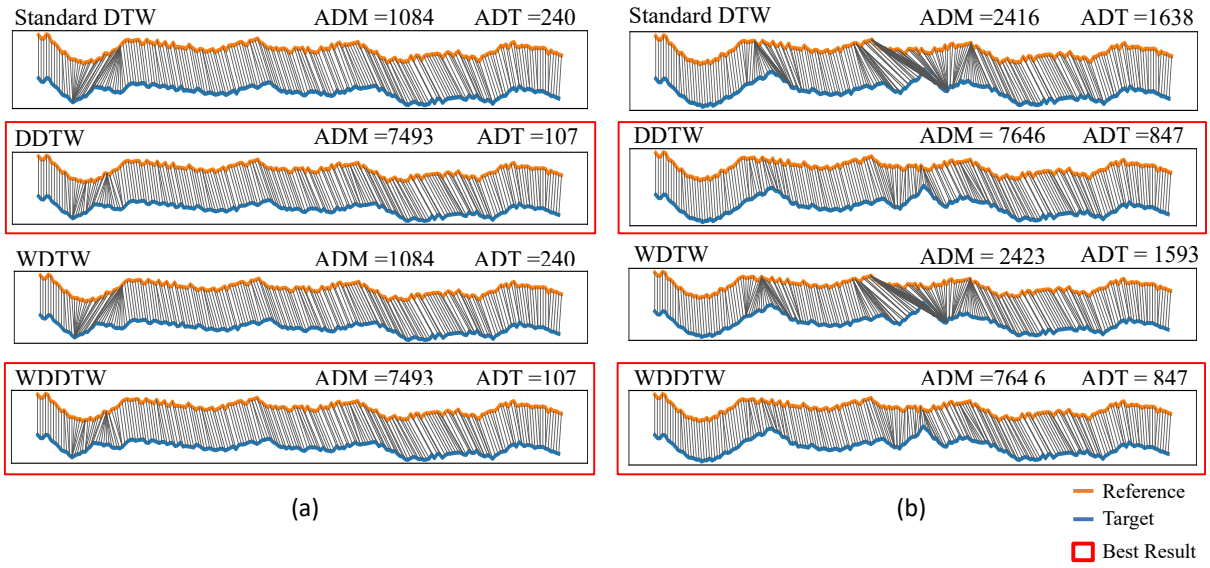

Figure 34: Alignment results from DTW variants when target contains both scaling and RGP as described in Fig-27. (a) Derivative variants of DTW perform well with single peak and scaled signal pairs over ADT measure. (b) WDDTW outperforms others on ADT over the signal pairs with scaling and multiple RGPs. While standard DTW is still the best on ADM measure in both instances. Note that axis labels are not shown here, as the visualization aims to show the alignment information.

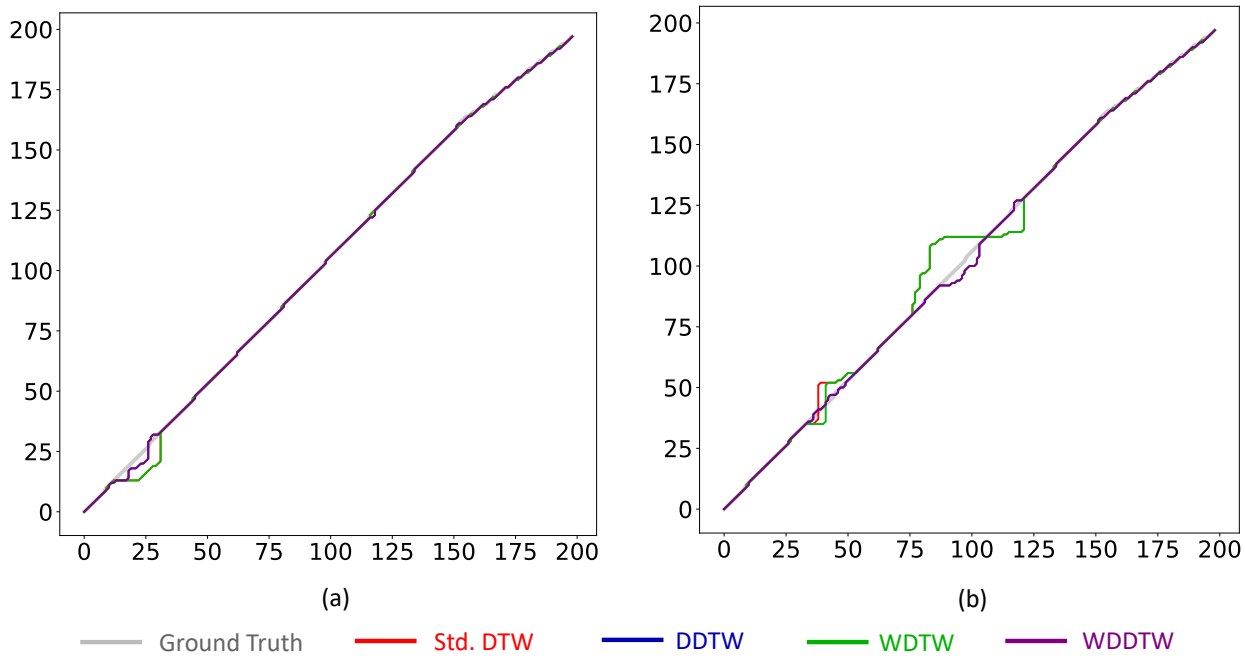

Figure 35: Warping path for the alignment shown in Fig-34 (a) and (b), respectively. (a) When target series has single RGP with scaling, all four DTW variants fail to align the region around the peak. (b) when we have target scaled with multiple RGP, none of the DTW variants is able to accurately align the pair, but the weighted DTWs are slightly closer to the true alignment than the non-weighted variants. x and y axes represent indices of the reference and target signals, respectively.

Our guidelines are based on the experimental evaluation of DTW measures shown in Table:Results and fig:Results$_hist$.

Table 1: Performance of different variations of DTW (average distance in m over 50 signal pairs for each variation.)

Signal Variation	Standard DTW ADM(ADT)	DDTW ADM(ADT)	WDTW ADM(ADT)	WDDTW ADM(ADT)
Scaled	183(93)	211(97)	860(331)	508(446)
Scaled but same size	851(126)	857(132)	2506(610)	1495(755)
RGP	707 (223)	196(41)	1050(4)	223(4)
MRGP	1920(768)	494(103)	3079(8)	563(10)
Scaled and RGP	1408(406)	956(178)	3377(638)	1620(769)
Scaled and MRGP	2731(1335)	1160(909)	5418(178)	1850(153)

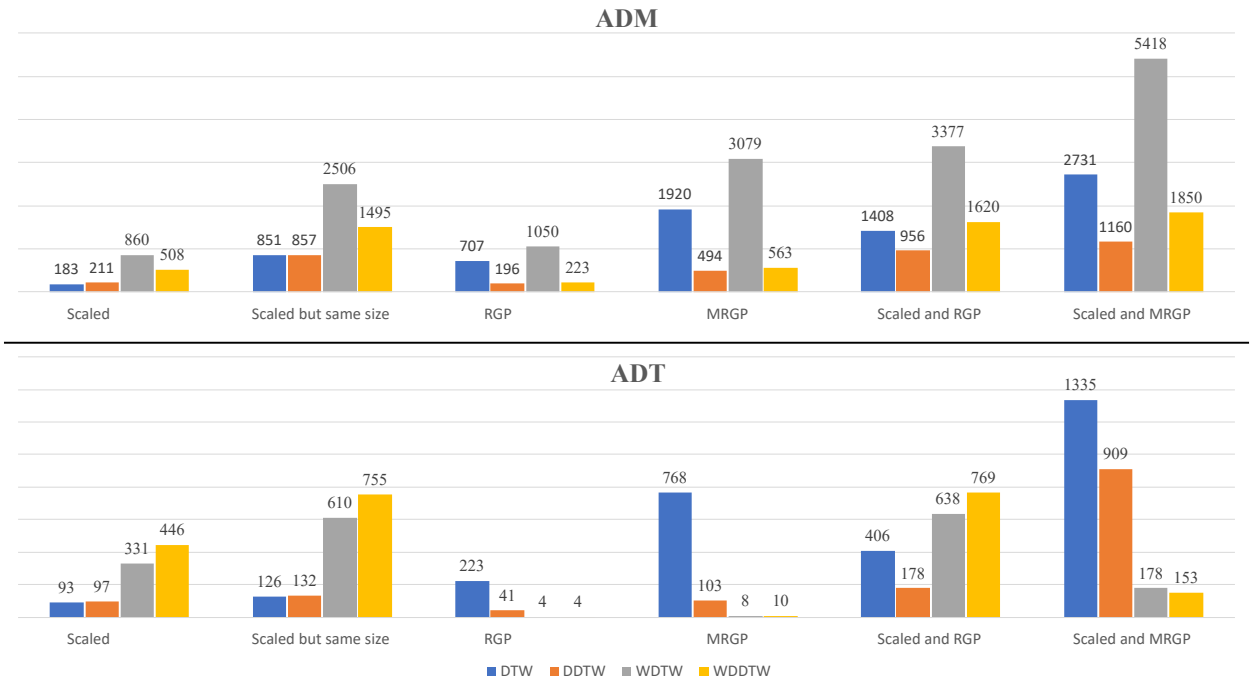

Figure 36: Performance of different variations of DTW. The average distance in 50 signal pairs for each variation is shown.

6.2 DTW Variants on Alignment of Surface Transitions with Gamma Ray Logs

As briefly mentioned in sec:background, DTW can be applied to the alignment of formation transitions with gamma ray logs. Gamma ray logs are usually recorded depth wise and it resembles linear time series data if we consider depth axis as time axis. The magnitude of gamma ray intensity provides information about different formations and their transitions at different depths.

When there are information available about the depth of formation transitions at certain location, the gamma ray logs of that location along with the depth information can be treated as reference. To know the depth of the formation transitions of the nearby locations, new gamma ray logs from these locations can be aligned with the reference.

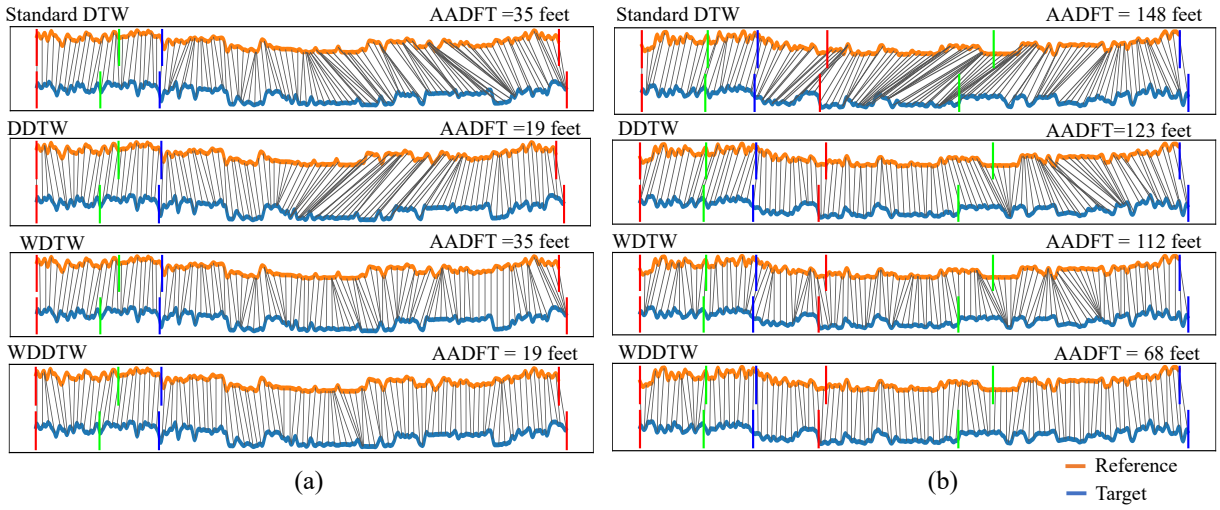

Figure 37: Alignment of Formation Transitions using gamma ray logs with different variants of DTW. (a) and (b) show two different pairs of gamma ray logs along with their formation transition marked with vertical bars (red, green and blue). The optimized weight parameters are {g=0.02 in (a) and g=0.03 in (b)} for WDTW and WDDTW, respectively. Note that axis labels are not shown here, as the visualization shows the alignment information.

For this type of alignment problems, a better evaluation measure is to find the mismatch in the alignment of important features. Here, we compute the aggregate absolute difference between

matched depth and true depth of formation transitions **(AADFT)**, which is similar to the previously introduced ADT measure. fig:Result$_SFT shows alignment of two instances of synthetic data which closely reser$

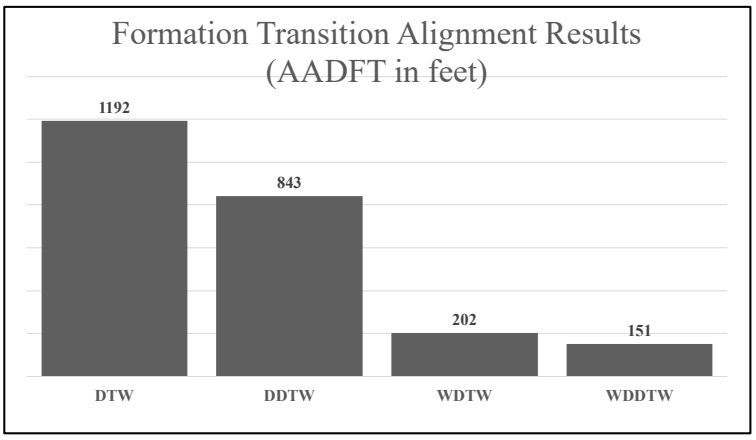

Figure 38: Average AADFT error for different DTW variants over 12 gamma ray signal pairs comprise of 6 formation transitions.

fig:SFT$_h ists shows the AADFT error value for each of the four DTW variants on 12 such pairs of logs.$**These gamn**

In order to find the type of difference/variance existing between two signals to-be-aligned, we can use our Simulated Annealing based parameter fitting framework discussed in sec:simulatedannealing. We record the parameters of each operation done by the framework while transforming a signal into the other. The effect of individual scaling as well as addition/subtraction of individual peaks performed by the framework are then evaluated in order to conclude if there exists significant amount of scaling or peak difference between the two signals. A peak has three parameters, that are magnitude of the center of the peak, width of the peak, and location of peak. Both magnitude and the width of the peak affect the alignment. The effect of a peak is directly proportional to the product of width of the peak and magnitude of the center. It can be mathematically quantified as,

$$Effect \propto (magnitude \times width)$$

[1]Due to the confidential nature of the data we cannot show the real data sequences

We need to normalize this with respect to the amplitude and length of the signal as,

$$Effect = \frac{magnitude \times width}{Amplitude \times length(Signal)} \times 100$$

For multiple iterations of the addition/subtraction of the peaks, the total effect is the sum of the effects created by individual peaks. If the total effect is larger or equal to 5%, then one of the two signals has non-trivial more peaks that do not exist in the other signal.

Similarly, the effect of scaling can be computed by simply considering the amount of change in length of the scaling window. We can normalize this change in the width of the scaling window with the length of the entire signal. If the sum of the absolute change in length is larger than 5%. then one of the signals has non-trivial featured scaled up/down as compared to the other.

The gamma ray log pairs shown in the fig:Result$_SFThave12\%and18\%differencesasscalingeffectand9\%and16$

7 Conclusion

In this thesis work, I have developed a new probability and envelope (layers) based visualization, called probabilistic envelope based technique (PE-technique), for the anomaly detection for time series data. This technique is capable of overcoming the known limitations of the existing methods. The PE-technique also provides quantitative information about the data as compared to the references, and capable of reducing the effect of outliers in the references. With slight modification it can be extended for the detection of over/under gauge in bore hole application. I presented the implementation and applications of this technique.

To address the lack of a guidance in feature alignment methods for time series data, I introduced a synthesis framework to generate signal pairs with known variation of features. The variation between a generated signal pair is user controllable. The framework generates a reference signal first and then deforms it to generate the target signal. Two basic modifications (i.e., scaling and the addition of Gaussian peaks) and their combination (i.e., both scaling and random peaks) are considered. To evaluate the effectiveness of this synthesis framework, parameter fitting is performed with the help of Simulated Annealing to transform a real-world signal to look like another. As presented, this framework is able to convert a signal into another with the help of multiple iterations of deformations. With the use of synthesis framework, the correct matching between a pair of signals is known. To utilize this known matching for a more accurate evaluation, a new distance measure for signal alignment is proposed, which is referred to as the Aggregate Distance over Time (ADT). ADT is based on the correct matching between a pair of signals. I performed extensive experiments for feature alignment on signal pairs with known variations using different variants of Dynamic Time Warping (DTW) methods. A detailed report of the results is provided along with the guidelines for which DTW variant works better with a particular variation being presented.

Limitation and Future Works Even though our experiments show that the PE-technique performs much better than the standard methods in the anomaly detection, it requires some modifications when applied to different tasks as shown in the application of over/under gauge detection. It is yet to assess how easily the proposed PE-technique can be adapted to other more challenging situations, which I plan to explore in the future. Another limitation of the PE-technique is that it is computed based on the probabilistic information. In the future, other statistics information can be used to aid in the design of the visualization to address different needs of specific applications.

The synthesis framework presented in this thesis proved to be able to transform a linear signal into another, but it takes many iterations of modifications. I suspect this is due to the use of only Gaussian peaks. Using different types of peaks along with Gaussian peaks may improve the efficiency of this framework. Also, our evaluation of alignment of different variations of the signals can be extended to application based evaluation with known variations. It can also be extended to other DTW variants like soft-DTW and multi-dimensional DTW. In addition, the evaluation can be extended to classification and clustering of time series data using different DTW variants. Furthermore, this work can be extended to evaluate different step patterns or different warping constraints on DTW variants as well as to evaluate open-ended DTW.

Bibliography

[1] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 49–54. IEEE, 2002.

[2] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[3] Xingyu Cai, Tingyang Xu, Jinfeng Yi, Junzhou Huang, and Sanguthevar Rajasekaran. Dtwnet: a dynamic time warping network. In *Advances in Neural Information Processing Systems*, pages 11636–11646, 2019.

[4] Lee W Campbell, David A Becker, Ali Azarbayejani, Aaron F Bobick, and Alex Pentland. Invariant features for 3-d gesture recognition. In *Proceedings of the second international conference on automatic face and gesture recognition*, pages 157–162. IEEE, 1996.

[5] Guoqing Chen, Qiang Wei, and Hong Zhang. Discovering similar time-series patterns with fuzzy clustering and dtw methods. In *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, volume 4, pages 2160–2164. IEEE, 2001.

[6] Mooi Choo Chuah and Fen Fu. Ecg anomaly detection via time series analysis. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 123–135. Springer, 2007.

[7] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 894–903. JMLR. org, 2017.

[8] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *KDD*, volume 98, pages 16–22, 1998.

[9] Hoang Anh Dau, Diego Furtado Silva, François Petitjean, Germain Forestier, Anthony Bagnall, Abdullah Mueen, and Eamonn Keogh. Optimizing dynamic time warping's window width for time series data mining applications. *Data Mining and Knowledge Discovery*, 32(4):1074–1120, 2018.

[10] Marcos Faundez-Zanuy. On-line signature recognition based on vq-dtw. *Pattern Recognition*, 40(3):981–992, 2007.

[11] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[12] C Godin and P Lockwood. Dtw schemes for continuous speech recognition: a unified view. *Computer Speech & Language*, 3(2):169–198, 1989.

[13] Francesco Gullo, Giovanni Ponti, Andrea Tagarelli, and Sergio Greco. A time series representation model for accurate and fast similarity detection. *Pattern Recognition*, 42(11):2998–3014, 2009.

[14] Tomasz Górecki and Maciej Łuczak. Multivariate time series classification with parametric derivative dynamic time warping. *Expert Systems with Applications*, 42(5):2305 – 2312, 2015.

[15] Bin Huang and W Kinsner. Ecg frame classification using dynamic time warping. In *IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings (Cat. No. 02CH37373)*, volume 2, pages 1105–1110. IEEE, 2002.

[16] Hesam Izakian and Witold Pedrycz. Anomaly detection in time series data using a fuzzy c-means clustering. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pages 1513–1518. IEEE, 2013.

[17] Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39:235–244, 2015.

[18] Young-Seon Jeong and Raja Jayaraman. Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification. *Knowledge-based Systems*, 75:184–191, 2015.

[19] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

[20] Uwe Jugel, Zbigniew Jerzak, Gregor Hackenbroich, and Volker Markl. M4: a visualization-oriented time series data aggregation. *Proceedings of the VLDB Endowment*, 7(10):797–808, 2014.

[21] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001.

[22] Marek Kulbacki and Artur Bak. Unsupervised learning motion models using dynamic time warping. In *Intelligent Information Systems 2002*, pages 217–226. Springer, 2002.

[23] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89. Presses universitaires de Louvain, 2015.

[24] Victor Maus, Gilberto Câmara, Ricardo Cartaxo, Alber Sanchez, Fernando M Ramos, and Gilberto R De Queiroz. A time-weighted dynamic time warping method for land-use and land-cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(8):3729–3739, 2016.

[25] Abdullah Mueen. Time series motif discovery: dimensions and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):152–159, 2014.

[26] Tim Oates, Laura Firoiu, and Paul R Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21. Citeseer, 1999.

[27] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[28] Toni M Rath and Raghavan Manmatha. Word image matching using dynamic time warping. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003.

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

[30] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[31] A Piyush Shanker and AN Rajagopalan. Off-line signature verification using dtw. *Pattern Recognition Letters*, 28(12):1407–1414, 2007.

[32] Jarke J Van Wijk and Edward R Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pages 4–9. IEEE, 1999.

[33] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Infovis*, volume 1, pages 7–14, 2001.

[34] Victor Wegner Maus, Gilberto Câmara, Marius Appel, and Edzer Pebesma. dtwsat: Time-weighted dynamic time warping for satellite image time series analysis in r. *Journal of Statistical Software*, 88(5):1–31, 2019.

[35] Xianglilan Zhang, Jiping Sun, and Zhigang Luo. One-against-all weighted dynamic time warping for language-independent and speaker-dependent speech recognition in adverse conditions. *PloS one*, 9(2), 2014.