

# GT-DTW: Bridging Graph Theory and Dynamic Time Warping for Complex Time Series Analysis

Sachit Mahajan

**Abstract**—Classification of time series data plays a critical role across various domains, enabling pattern recognition and trend prediction. Traditional methods like Dynamic Time Warping (DTW) have been widely used to measure similarity between time series, but there are challenges related to computational complexity and sensitivity to noise. The conventional DTW approach, with its quadratic time complexity, can be inefficient for large datasets, and some implementations may struggle with noise and local variations. To overcome these limitations, a novel method called Graph-Theoretic Dynamic Time Warping (GT-DTW) is proposed. GT-DTW represents each time series as a graph, applies DTW on the graph representations and calculates the distances between different time series based on these graph representations. This approach provides a robust and computationally efficient method for time series classification, and experimental results show that GT-DTW provides better results when compared with conventional methods on the benchmark datasets from the UCR time series database. GT-DTW also demonstrates enhanced effectiveness in situations where time series share fundamental similarities, yet are affected by intricate transformations, noise, inconsistencies in length, and localized distortions.

**Index Terms**—classification, time series, dynamic time warping, graph theory

## I. INTRODUCTION

The exponential proliferation of Big Data, primarily propelled by the advent of the Internet of Things (IoT), sensor networks, and state-of-the-art high-frequency data collection technologies, has precipitated a remarkable surge in the generation of time series data across a broad spectrum of domains [1]–[5]. From finance and healthcare to telecommunication, climatology, environmental monitoring and social sciences, the utility and potential of time series data for informing decision-making processes and providing valuable insights is an aspect that is increasingly gaining recognition [6]–[8]. However, the analysis and interpretation of time series data are not without challenges. The unprecedented volume and intricate complexity associated with time series data pose significant hurdles that need to be overcome to fully exploit its potential.

Historically, time series analysis has been utilized with a focus on discerning patterns [9]–[11], projecting future trends [5], and explaining latent structures inherent within the data [12]. Nevertheless, conventional analysis methodologies, predominantly reliant on the time domain for operation, often falter when faced with the inherent intricacy of time series

data. The underlying complexity stems from a variety of factors, including high dimensionality, non-stationarity, noise, and the imperative of acknowledging temporal dependencies.

Time-series classification (TSC) [13], a crucial aspect of time series analysis, has been traditionally achieved by employing techniques that operate in the time domain. The Euclidean distance measure, being the simplest and most intuitive, computes the straight-line distance between two points in a multidimensional space [14]. Despite its simplicity, the Euclidean distance is acutely susceptible to distortions in the time axis, and it fails to encapsulate the temporal dependencies between distinct segments of the series [15].

In an effort to address the limitations associated with time domain-based techniques, the Dynamic Time Warping (DTW) [16] method has been widely adopted in TSC [17]. DTW quantifies the similarity between two temporal sequences, notwithstanding disparities in their speed or length, thereby efficiently tackling distortions in the time axis. Nevertheless, despite its improvement over the Euclidean distance method, DTW is not without its drawbacks. It continues to operate within the time domain, and it exhibits high computational demands [18], especially when handling large-scale datasets. This underscores an urgent need for the evolution and exploration of alternative, more efficient techniques for TSC that are capable of adeptly managing the convoluted characteristics often observed in time series data.

A promising avenue for improvement emerges from the field of graph theory, a mathematical discipline devoted to investigating the relationships between entities. This theory has been incorporated into time-series analysis with promising results [19]. By translating time-series data into graph structures, intricate structures and patterns embedded in the data can be discerned more effectively, and temporal dependencies within the series can be taken into account. However, the effectiveness of graph-based methods often hinges on predefined graph structures, which may not fully encapsulate the complexity inherent to time-series data.

Although significant advancements have been made in the field of time series analysis, a substantial gap still exists between its utility and performance. This study seeks to bridge this gap with the introduction of a novel method, Graph-Theoretic Dynamic Time Warping (GT-DTW), a novel method that combines the flexibility of DTW with the representational power of graph theory. GT-DTW represents each time series as a graph, where nodes represent segments of the time series and edges represent the temporal dependencies between them. By applying DTW on these graph representations, an effective comparison of time series based on their topological properties

Sachit Mahajan is with the Professorship of Computational Social Science, ETH Zurich, Rämistrasse 101, 8092 Zurich Switzerland. Email: sachit.mahajan@gess.ethz.ch

Manuscript received ; revised

can be achieved.

GT-DTW provides several advantages. First, by representing time series as graphs, complex structures and patterns in the data can be captured. Second, by applying DTW on these graph representations, time series that have different lengths or are out of phase can be effectively compared. Finally, GT-DTW is designed to be computationally efficient, making it suitable for large-scale datasets.

The contributions of this work are two-fold:

- This work introduces GT-DTW, a novel approach for time-series classification that extends the principles of DTW and incorporates the structural and temporal insights provided by graph theory. This innovative approach addresses the limitations of traditional DTW and offers a more robust and accurate classification of time-series data. The integration of graph theory principles helps in handling complex scenarios, such as differing lengths and inherent temporal dependencies in time-series data.
- An extensive evaluation of GT-DTW is conducted on both benchmark datasets and complex simulated scenarios, comparing its performance with other conventional methods. This rigorous evaluation demonstrates the effectiveness of GT-DTW in handling the challenges of time-series classification and showcases its potential for wider application.

The rest of this paper is structured as follows: Section II reviews the existing literature in the field of time-series classification. It highlights the limitations of current techniques and underscores the need for innovative approaches like GT-DTW. Section III provides a detailed explanation of the GT-DTW methodology. Section IV presents the results from applying GT-DTW on different datasets and compares its performance with other methods, offering an empirical evaluation of GT-DTW's effectiveness. Finally, Section V discusses the implications of the findings, potential applications of GT-DTW, directions for future research, and concludes the paper.

## II. RELATED WORK

The field of time-series analysis is a dynamic and complex landscape, with a multitude of techniques and models that have been developed over the years. The central focus of this section is to delve into the existing literature on time-series processing and classification, highlighting the key challenges associated with these tasks, and offering an in-depth look into the existing models such as Euclidean distance measure, DTW, and its variants.

### A. Time-series Processing and Classification

The processing and classification of time-series data is a widely researched domain with profound implications across various disciplines. However, several factors make these tasks quite challenging. The intrinsic characteristics of time-series data such as high dimensionality, non-stationarity, noise, and temporal dependencies present significant hurdles [20]. Additionally, time-series data often have a high degree of similarity, making class distinctions ambiguous, and thus posing a challenge to classification tasks [21].

### B. Euclidean Distance Measure

One of the most basic and easy-to-understand techniques used in time-series classification is the Euclidean distance measure. This method calculates the shortest distance between two points in a space with multiple dimensions. Although it is a simple and effective method, there are some significant limitations. It is very susceptible to distortions in the time axis and does not effectively capture the connections between different parts of the series over time [22].

### C. Dynamic Time Warping (DTW)

DTW, first introduced by Sakoe and Chiba [23], emerged as a more robust alternative to the Euclidean distance measure for time-series classification. DTW measures the similarity between two temporal sequences, accommodating variations in speed and length and thereby effectively addressing distortions in the time axis. Despite its strengths, it still operates within the time domain and incurs significant computational overhead, making it less efficient for large-scale datasets [24].

Several variations of DTW have been proposed in the literature to overcome its inherent drawbacks [25]. These include FastDTW, a more efficient implementation of DTW proposed by Salvador and Chan [26], which reduces the computational complexity through an approximate solution. Another variation known as the Derivative DTW is a modification of DTW that uses the derivative of the time series to calculate the distance between two points. This reduces the computational complexity of the algorithm while still capturing the shape of the time series [27]. Similarly, the Weighted DTW [28] introduces a local weighting scheme into the DTW algorithm to enhance its discriminative power.

Despite the significant advancements made in the DTW algorithm, there still exist considerable challenges. A recurrent issue with the conventional DTW algorithm is that it is computationally expensive, especially when dealing with long time series data [29]. This is because the DTW algorithm requires computing the distance between each pair of points in the two time series, which results in a time complexity [30]–[32]. Also, traditional DTW algorithm [33] is usually ineffective in handling many-to-one or one-to-many instances, wherein the number of points in the contrasting time series can be extremely disparate. Furthermore, it fails to adeptly manage the concern of time series of unequal lengths, a frequent phenomenon encountered in practical applications. Also, several refined DTW algorithms necessitate manually preselected parameters, potentially constraining their versatility across diverse datasets [31]. These shortcomings can compromise the accuracy of classification results and constrict the broader applicability of these methods.

### D. Deep Learning Approaches

Deep learning has become a prominent technique in TSC due to its ability to automatically learn features from raw data, thus bypassing the necessity for manual feature extraction [34]. Various deep learning architectures have been proposed for time-series classification tasks, including Convolutional

Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer models.

LSTM networks, a specific type of Recurrent Neural Networks (RNNs), have gained attention in TSC owing to their capacity to capture long-term dependencies in the data [35]. Unlike traditional methods, LSTMs can model complex temporal relationships and are less sensitive to the sequence length or temporal shifts in the data. However, they come with high computation complexity [36].

The LSTM architecture includes memory cells that allow the network to store and retrieve information over long sequences, making it particularly suited for time-series data [37]. Despite their efficacy, deep learning models are not devoid of challenges. They are sensitive to hyperparameter settings and may require extensive tuning, especially in scenarios where the data is imbalanced or the distribution is complex [38]. Other architectures like CNNs and Transformers have also been adapted for TSC [39], [40]. CNNs exploit spatial hierarchies through their convolutional layers, while Transformers employ self-attention mechanisms to capture temporal dependencies. Both architectures have demonstrated competitive performance in various benchmarks but share the common drawbacks of requiring large datasets and computational resources.

This brief overview of the related works emphasizes the importance and challenges of time-series analysis. The ongoing efforts for efficient and accurate time-series classification techniques forms the basis for the proposed GT-DTW method. The subsequent sections of this paper delve into the details of this method, discussing its conceptual underpinnings and demonstrating its efficacy.

### III. METHODOLOGY

This section goes over the GT-DTW approach to time series classification in further detail. The following subsections provide an in-depth look into each of the key components of GT-DTW methodology.

#### A. Dataset Normalization

Normalization is an essential preprocessing step in machine learning and data analysis, aimed at bringing the feature values onto a common scale without distorting the differences in the range of values. This is of particular importance when dealing with high-dimensional data such as time series, where the range of values can vary significantly across dimensions. In this work, Z-score normalization is used, a technique that rescales the features to have a mean ( $\mu$ ) of 0 and a standard deviation ( $\sigma$ ) of 1, thereby ensuring that the features follow a standard normal distribution. Given a feature  $x$ , its normalized value  $x'$  is calculated as follows:

$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

Z-score normalization was chosen due to its robustness to outliers [41] and its ability to standardize features regardless of their original distribution, making it suitable for time series data.

#### B. Feature Extraction

Feature extraction is the process of transforming raw data into a set of features (or a feature vector) that can be effectively used in machine learning algorithms. In time series analysis, this often involves converting the time series data from the time domain to the frequency domain, which can reveal important patterns and properties of the data. The Fourier Transform is used for this purpose, as it efficiently decomposes a time-domain function (a signal) into its constituent frequencies [42]. This transformation is particularly beneficial for this study because it allows the capture of the essential frequency characteristics of the time series, revealing patterns and structures that are not readily apparent in the time domain. Specifically, the first half of the Fourier Transform is used as features for each time series, denoted as  $F(t)$ . This choice is based on the symmetric properties of the Fourier Transform for real-valued time series, where the second half of the transformed signal is a complex conjugate mirror image of the first half. Since the frequency information is entirely captured in the first half of the transform, using only this part can reduce redundancy, reduce computational complexity, and focus on the most relevant frequency components that describe the underlying structure of the time series [43]. The Fourier Transform  $F(t)$  of a time series  $t$  is given by:

$$F(t) = \int_{-\infty}^{\infty} t(x)e^{-2\pi ix} dx \quad (2)$$

where  $t(x)$  is the time series,  $e$  is the base of the natural logarithm,  $i$  is the imaginary unit, and  $x$  is the frequency variable.

#### C. Creating the Gaussian Mixture Model (GMM)

After extracting the features, a Gaussian Mixture Model (GMM) [44] is used to represent each time series. GMMs are probabilistic models that assume all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. They provide a flexible and powerful tool for representing different “modes” or clusters within the data. The use of GMMs in this context allows us to capture complex, multimodal distributions of the time series data, thereby providing a more accurate and informative representation of the time series.

Given a set of features  $X = \{x_1, x_2, \dots, x_n\}$ , the GMM aims to maximize the log-likelihood function:

$$\log p(X|\lambda) = \sum_{i=1}^n \log \left[ \sum_{j=1}^m w_j \phi(x_i|\mu_j, \Sigma_j) \right] \quad (3)$$

where  $m$  is the number of mixtures,  $w_j$  is the weight of the  $j^{th}$  component,  $\phi(x_i|\mu_j, \Sigma_j)$  is the Gaussian distribution with mean  $\mu_j$  and covariance  $\Sigma_j$ , and  $\lambda = \{w_j, \mu_j, \Sigma_j\}_{j=1}^m$  are the model parameters.

#### D. Graph Creation

Next, each GMM is represented as a graph, where the nodes correspond to the components of the GMM, and the edges

represent the distance between the components. The use of a graph-based representation allows us to capture the inherent structure and dependencies within the time series data. This is accomplished through the use of a directed graph  $G(V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. Each node  $v_i \in V$  corresponds to a GMM component, and an edge  $e_{ij} \in E$  represents the distance between components  $v_i$  and  $v_j$ .

The distance between nodes is computed using the  $k$ -nearest neighbors algorithm, a popular method for graph construction due to its simplicity and effectiveness. For each node  $v_i$ , the Euclidean distance to every other node is calculated, and it is connected to its  $k$  nearest nodes. The choice of a directed graph and the use of the  $k$ -nearest neighbors algorithm allow us to capture the inherent directionality and temporal dependencies within the time series.

#### E. Cost Matrix Calculation

The cost matrix is a crucial component of the GT-DTW algorithm. It represents the dissimilarity between each pair of nodes (components) in the two graphs. Given two graphs  $G_1$  and  $G_2$  with  $n_1$  and  $n_2$  nodes respectively, the cost matrix  $C$  is an  $n_1 \times n_2$  matrix where each element  $c_{ij}$  represents the dissimilarity between node  $i$  in  $G_1$  and node  $j$  in  $G_2$ . The dissimilarity is calculated using the Manhattan distance (L1 norm) between the mean vectors of the corresponding GMM components:

$$c_{ij} = \|\mu_{1i} - \mu_{2j}\| \quad (4)$$

where  $\mu_{1i}$  and  $\mu_{2j}$  are the means of the GMM components corresponding to node  $i$  in  $G_1$  and node  $j$  in  $G_2$  respectively. The Manhattan distance was chosen due to its robustness [45] to high-dimensional data and its suitability for comparing distributions, which aligns well with GMM-based representation.

#### F. Graph-Theoretic Dynamic Time Warping (GT-DTW)

The next core component of the proposed method is the application of the DTW algorithm to the cost matrix. DTW is a well-established dynamic programming algorithm that finds an optimal alignment between two sequences of numeric values. In this case, these sequences are the rows of the cost matrix. The GT-DTW distance between the two graphs is the total cost of this optimal alignment.

Let  $D$  be the cumulative distance matrix, the same size as the cost matrix  $C$ . Each element  $d_{ij}$  of  $D$  is calculated as follows:

$$d_{ij} = c_{ij} + \min\{d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}\} \quad (5)$$

where  $c_{ij}$  is the cost matrix element and  $\min\{d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}\}$  represents the minimum cost to reach this point. The GT-DTW distance is then the value in the bottom right corner of the cumulative distance matrix  $D$ , i.e.,  $d_{n_1, n_2}$ .

The application of DTW in this context allows us to take into account the topological properties of the time series, represented by the graph structure, in addition to the raw time

TABLE I  
SUMMARY OF DATASETS.

Dataset	Data Type	Number of Classes	Train Size	Test Size	Time Series Length
Trace	Sensor	4	100	100	275
GunPoint	Motion	2	50	150	150
Fish	Image	7	175	175	463
ECG200	ECG	2	100	100	98
Beef	Spectro	5	30	30	470
FaceAll	Image	14	560	1690	131
FaceFour	Image	4	24	88	350
Lightning2	Sensor	2	60	61	637
OliveOil	Spectro	4	30	30	570
OSULeaf	Image	6	200	242	427

series data. This results in a more robust and accurate measure of time series similarity.

#### G. Hyperparameter Tuning

The proposed method requires tuning of two hyperparameters:  $n_{components}$ , which is the number of components in the GMM, and  $k$ , the number of nearest neighbors in the graph. To determine the optimal values for these hyperparameters, a grid search strategy is used [46]. In this approach, an iteration is performed over all possible combinations of hyperparameters within a predefined range. For each combination, the classification accuracy of the model on the test data is calculated. The combination that results in the highest accuracy is chosen as the optimal set of hyperparameters. The use of grid search ensures a comprehensive exploration of the hyperparameter space and reduces the risk of missing the optimal configuration due to random chance.

#### H. Classification

After the GT-DTW distances between all pairs of time series in the training and test sets have been computed, these distances are used to classify the test time series. Specifically, a Random Forest Classifier is used, a robust machine learning model that can handle non-linear data and avoid overfitting by averaging the predictions of several decision trees. Random Forest Classifier was chosen due to its proven performance across a wide range of classification tasks and its robustness to overfitting, making it a suitable choice for time-series classification task [47].

Given a set of training instances  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , where  $x_i$  is the feature vector and  $y_i$  is the corresponding label, and a test instance  $x$ , the Random Forest Classifier generates  $B$  decision trees. Each tree  $b$  provides a prediction  $y_b(x)$ . The final prediction  $y(x)$  is the majority vote of all predictions:

$$y(x) = \arg \max_y \sum_{b=1}^B I(y_b(x) = y) \quad (6)$$

where  $I(\cdot)$  is the indicator function.

TABLE II  
PERFORMANCE EVALUATION OF GT-DTW, FASTDTW, EUCLIDEAN, AND LSTM METHODS ON SELECTED DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Dataset	GT-DTW		FastDTW		Euclidean		LSTM	
	Accuracy	Error Rate	Accuracy	Error Rate	Accuracy	Error Rate	Accuracy	Error Rate
Trace	<b>0.81</b>	<b>0.19</b>	0.75	0.25	0.76	0.24	0.54	0.46
GunPoint	<b>0.91</b>	<b>0.09</b>	0.89	0.11	<b>0.91</b>	<b>0.09</b>	0.76	0.24
Fish	0.76	0.24	<b>0.81</b>	<b>0.19</b>	0.77	0.23	0.30	<b>0.70</b>
ECG200	0.81	0.19	<b>0.83</b>	<b>0.17</b>	<b>0.83</b>	<b>0.17</b>	0.76	0.24
Beef	<b>0.73</b>	<b>0.27</b>	<b>0.73</b>	<b>0.27</b>	0.67	0.33	0.33	0.67
FaceAll	<b>0.78</b>	<b>0.22</b>	0.77	0.23	0.73	0.27	0.58	0.42
FaceFour	<b>0.76</b>	<b>0.24</b>	0.71	0.29	0.71	0.29	0.43	0.57
Lightning2	<b>0.72</b>	<b>0.28</b>	0.69	0.31	0.71	0.29	0.62	0.38
OliveOil	<b>0.93</b>	<b>0.07</b>	0.90	0.10	0.90	0.10	0.57	0.43
OSULeaf	<b>0.51</b>	<b>0.49</b>	0.47	0.53	0.46	0.54	<b>0.51</b>	<b>0.49</b>

#### IV. RESULTS

##### A. Evaluation

We evaluated four time series classification methods: GT-DTW, FastDTW, Euclidean distance, and LSTM. GT-DTW, our proposed method, employs Gaussian Mixture Models and graph theory to improve upon DTW. FastDTW serves as a computationally efficient approximation to standard DTW. Euclidean distance provides a straightforward measure of similarity between time series. A single layer LSTM with 50 units is used to capture long-term dependencies and utilizes a softmax output layer for multi-class classification. These methods were applied to ten diverse datasets sourced from the UCR time series classification archive [48]. The datasets, namely Trace, GunPoint, Fish, ECG200, Beef, FaceAll, FaceFour, Lightning2, OliveOil, and OSULeaf, were carefully chosen to represent a wide variety of time series data, encompassing fields such as motion, electrocardiogram signals, images, sensor data, and spectrography. Table I provides an overview of the datasets used in this study.

Each method's performance was evaluated using five metrics: accuracy, error rate, precision, recall, and F1 score. The results provide insights into the performance of each method across different types of datasets and serve as a benchmark for further research in time series classification. Classification accuracy is the fraction of correct predictions made by the model out of all predictions. Given the total number of predictions  $N$  and the number of correct predictions  $C$ , the accuracy can be defined as:

$$\text{Accuracy} = \frac{C}{N} \quad (7)$$

The error rate is the complement of the accuracy and represents the fraction of incorrect predictions. It can be defined as:

$$\text{Error Rate} = 1 - \text{Accuracy} \quad (8)$$

Table II presents a comprehensive performance evaluation of four time-series classification methods: GT-DTW, FastDTW, Euclidean, and LSTM. The table provides key insights into the efficacy and robustness of these methods across different datasets.

Firstly, GT-DTW consistently outperformed the other methods in terms of accuracy and error rate across multiple datasets. Specifically, GT-DTW demonstrated superior performance in

the Trace, GunPoint, Beef, FaceAll, FaceFour, Lightning2, OliveOil, and OSULeaf datasets. This indicates that GT-DTW is highly effective in capturing intricate temporal dynamics and is robustness across diverse data types. Secondly, the low error rates recorded for GT-DTW in the majority of the datasets reinforce its efficacy. This not only confirms the method's ability to make accurate predictions but also suggests its resilience to overfitting, thereby ensuring reliable and consistent performance.

In contrast, FastDTW and Euclidean methods also showed competitive performance but were generally less consistent. Notably, FastDTW outperformed GT-DTW in terms of accuracy in the Fish and ECG200 datasets, revealing that under certain conditions, FastDTW can be equally effective. The LSTM method produced mixed results. While it matched the performance of GT-DTW in the OSULeaf dataset, it generally performed poorly on the remaining datasets, particularly in terms of accuracy.

In addition to accuracy and error rates, the methods were also evaluated based on precision, recall, and F1 scores, as shown in Figure 1. Precision measures the proportion of true positive identifications in the dataset, recall refers to the proportion of actual positives that were identified correctly, and the F1 score is the harmonic mean of precision and recall.

Across the datasets, GT-DTW generally exhibited higher precision compared to FastDTW, Euclidean, and LSTM methods (Figure 1a). However, the difference in precision was marginal in some cases, indicating that GT-DTW, FastDTW, and Euclidean methods were equally adept at avoiding false positive identifications. LSTM, on the other hand, often showed lower precision, particularly in datasets like 'Fish' and 'Beef'.

Recall rates varied across the datasets and methods (Figure 1b). In some datasets, such as 'Trace', GT-DTW demonstrated superior recall rates. In others like 'OliveOil', GT-DTW, FastDTW, and Euclidean methods exhibited similar recall performance, while LSTM lagged behind. Regarding F1 scores, which consider both the precision and the recall, GT-DTW generally outperformed the other methods across most datasets, revealing a better balance between precision and recall (Figure 1c). This consistent performance highlights the effectiveness of GT-DTW in comparison with FastDTW, Euclidean, and LSTM methods across various time series classification tasks.

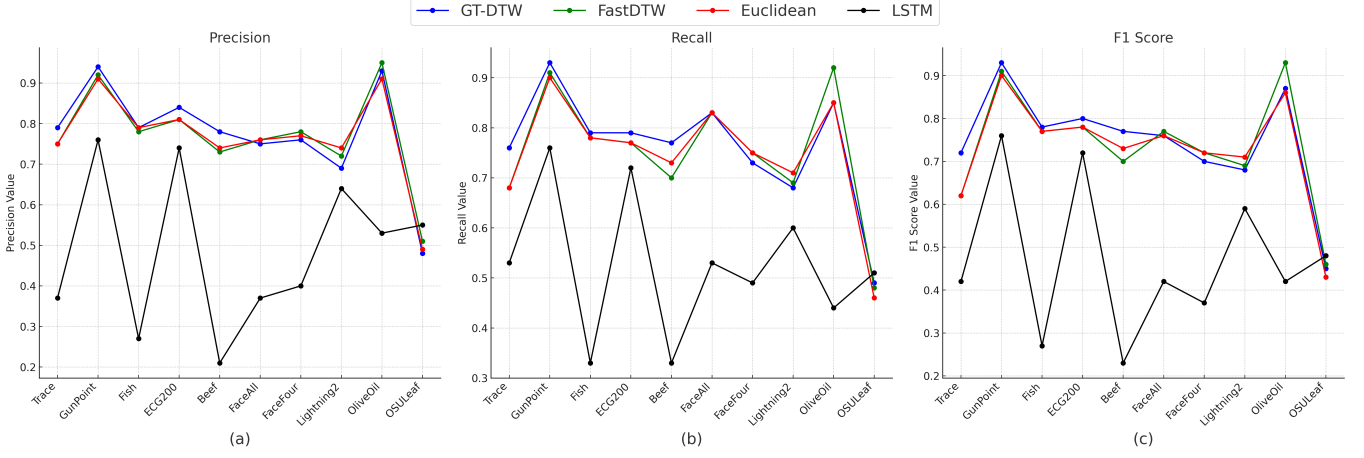


Fig. 1. Comparison of Classification Methods Across Various Datasets: This figure shows the performance of four different time-series classification methods—GT-DTW (blue), FastDTW (green), Euclidean Distance (red), and LSTM (black)—across multiple datasets. Sub-figures (a), (b), and (c) represent the Precision, Recall, and F1 Score metrics, respectively. The x-axis lists the different datasets, while the y-axis shows the metric values.

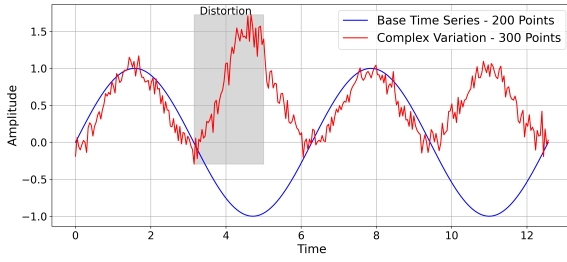


Fig. 2. A visual representation of the base time series (blue) and its complex variation (red). The variation has been crafted with non-linear transformations, noise, and local distortions to challenge traditional alignment methods.

The consistent performance of the GT-DTW method across a wide range of datasets signifies its efficiency and adaptability, making it a suitable choice for deployment in diverse applications. This adaptability could be particularly valuable in real-world situations where the data's consistency and quality might differ significantly.

### B. Understanding the Robustness of GT-DTW Through Synthetic Data Analysis

To further analyze the efficacy of the GT-DTW method, a synthetic data experiment was conducted. This experiment aimed to understand how GT-DTW fares against traditional DTW in scenarios where time series have inherent similarities but are presented with complex transformations, noise, different lengths and local distortions.

As depicted in Figure 2, the base time series is a simple sine curve and there is another variation of the base time series. The idea was to create a time series that, while fundamentally similar to the base, presents complexities that might challenge traditional alignment methods. It was created by applying a non-linear transformation (squaring the sine values), adding random noise to induce jitter, and introducing a local distortion between indices corresponding to time values approximately

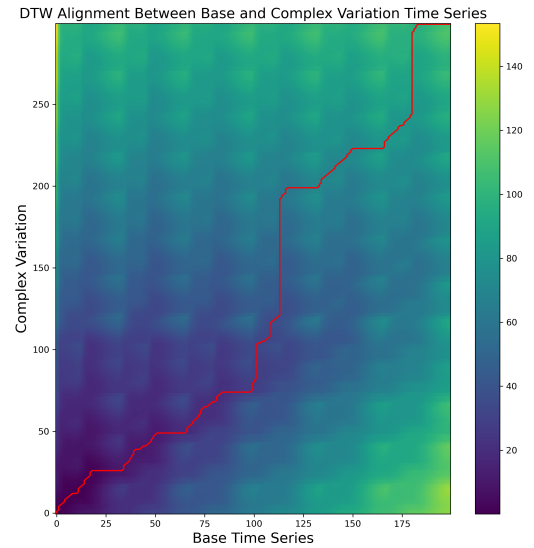


Fig. 3. Visualizations of alignments between the base time series and its complex variation using traditional DTW.

3.14 and 5.03. Furthermore, the variation time series was created with a different length (300 points) compared to the base time series (200 points) to simulate a many-to-one scenario.

a) *Interpreting the DTW and GT-DTW Heatmaps:* Both heatmaps (Figures 3 and 4) represent cost matrices. The cost matrix, in the context of DTW, is a two-dimensional array where each cell  $(i, j)$  represents the distance (or difference) between point  $i$  of the first time series and point  $j$  of the second time series.

The color intensity in the heatmap corresponds to the magnitude of this distance. Lighter colors (towards the yellow end of the spectrum) indicate higher distances (or greater

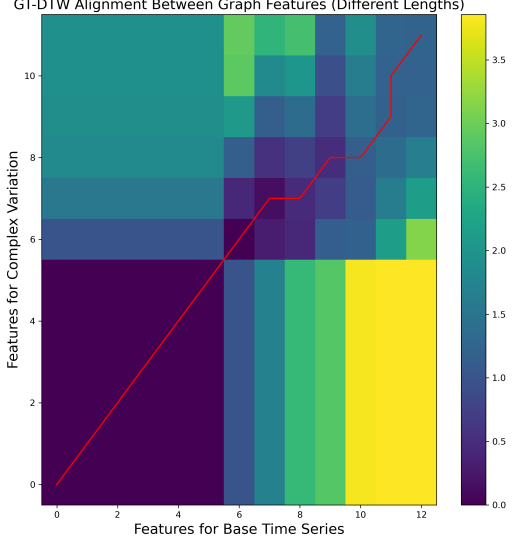


Fig. 4. Visualizations of alignments between the base time series and its complex variation using GT-DTW.

differences), while darker colors (towards the purple end of the spectrum) represent smaller distances (or similarities).

The red path superimposed on the heatmap is the optimal alignment path. This path represents the best way to align the two time series in a manner that minimizes the cumulative distance. In essence, it's the path through the cost matrix that results in the least overall difference between the two time series.

A few insights can be gathered from these heatmaps:

- A perfectly diagonal path suggests a one-to-one alignment between the two time series, meaning they are quite similar.
- Deviations from the diagonal indicate stretches where one segment of a time series aligns with multiple segments of the other, revealing discrepancies between the series.
- Areas of the heatmap that are consistently darker indicate regions of similarity between the two time series.
- Conversely, lighter regions highlight differences or disparities.

Upon applying traditional DTW (Figure 3), the alignment copes with the complexities of aligning the base and variation time series, especially considering the local distortion and different lengths. The alignment path, while predominantly diagonal, shows areas of deviation, signifying the challenges faced by traditional DTW in handling the non-linear transformation and noise.

On the other hand, the GT-DTW method (Figure 4) provides a more nuanced and consistent alignment. This alignment, premised on the graph-based features of the time series, adeptly captures the underlying structural similarities. The graph structures were built using GMMs and nearest neighbors, and features such as node degrees and clustering coefficients were extracted to represent the complexity of

the time series. The GT-DTW heatmap illustrates a more consistently diagonal path, symbolizing its effectiveness in recognizing structural similarities, even amidst the introduced complexities. This shows GT-DTW's robustness in handling the many-to-one scenario, where the lengths of the time series differ, and the complexity due to non-linear transformations, random noise, and local distortion.

To summarise, the GT-DTW method offers a more robust and insightful comparison by focusing on underlying graph-based structures rather than individual data points. While traditional DTW exhibits sensitivity to the transformations applied, GT-DTW's ability to encapsulate complex patterns and dependencies in the data makes it an invaluable tool for time series analysis in noisy and intricate scenarios. By handling different lengths and complex structures, GT-DTW demonstrates its potential as a powerful extension to traditional time series alignment techniques, bridging the gap between raw data and higher-level structural interpretation.

## V. CONCLUSION

The classification of time series data is important across various domains. Traditional methods like DTW often struggle with the complexities of time series data, particularly when dealing with complex temporal dependencies and varying lengths. To overcome these challenges, this paper has introduced GT-DTW, a novel technique that transcends conventional time series representation by portraying each time series as a graph. Through the innovative application of DTW on these graph representations, GT-DTW emerges as a robust and computationally efficient method for time series classification. Comparative analyses with FastDTW and Euclidean methods have demonstrated the efficacy of GT-DTW, highlighting its consistent performance across multiple datasets. While there were instances where other methods matched or slightly outperformed GT-DTW, the overall trend indicates the superior performance of GT-DTW in balancing precision, recall, and F1 scores. The detailed examination of GT-DTW's resilience to complexities such as non-linear transformations, random noise, and local distortions further substantiates its robustness. Its ability to comprehend complex structures in time series adds a layer of depth to its applicability. The GT-DTW method's efficiency and generalizability position it as a promising tool for varied applications, with significant implications in real-world scenarios where data consistency and quality may vary substantially.

Although the primary focus of this study is on the classification of time series data using the GT-DTW method, the underlying approach's versatility extends beyond mere classification. The graph-based representation and distance computation inherent in GT-DTW can be adapted to other analytical tasks such as similarity analysis, outlier detection, and clustering.

## REFERENCES

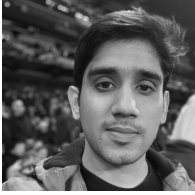
- [1] Xudong Guan, Chong Huang, Gaochuan Liu, Xuelian Meng, and Qing-sheng Liu. Mapping rice cropping systems in vietnam using an ndvi-based time-series similarity measurement based on dtw distance. *Remote Sensing*, 8(1):19, 2016.



- [2] Scott L Zeger, Duncan Thomas, Francesca Dominici, Jonathan M Samet, Joel Schwartz, Douglas Dockery, and Aaron Cohen. Exposure measurement error in time-series studies of air pollution: concepts and consequences. *Environmental health perspectives*, 108(5):419–426, 2000.
- [3] Sachit Mahajan. Design and development of an open-source framework for citizen-centric environmental monitoring and data analysis. *Scientific Reports*, 12(1):14416, 2022.
- [4] Ji Zhang, Hongzhou Li, Qigang Gao, Hai Wang, and Yonglong Luo. Detecting anomalies from big network traffic data using an adaptive detection approach. *Information Sciences*, 318:91–110, 2015.
- [5] Sachit Mahajan, Hao-Min Liu, Tzu-Chieh Tsai, and Ling-Jyh Chen. Improving the accuracy and efficiency of pm2. 5 forecast service using cluster-based hybrid neural network model. *IEEE Access*, 6:19193–19204, 2018.
- [6] Roonak Rezvani, Payam Barnaghi, and Shirin Enshaieifar. A new pattern representation method for time-series data. *IEEE Transactions on Knowledge and Data Engineering*, 33(7):2818–2832, 2019.
- [7] Ruey S Tsay. *Analysis of financial time series*. John Wiley & sons, 2005.
- [8] Cyuan-Heng Luo, Hsuan Yang, Li-Pang Huang, Sachit Mahajan, and Ling-Jyh Chen. A fast pm2. 5 forecast approach based on time-series data analysis, regression and regularization. In *2018 conference on technologies and applications of artificial intelligence (TAAI)*, pages 78–81. IEEE, 2018.
- [9] Sachit Mahajan and Jenny Martinez. Water, water, but not everywhere: Analysis of shrinking water bodies using open access satellite data. *International Journal of Sustainable Development & World Ecology*, 28(4):326–338, 2021.
- [10] Ling-Jyh Chen, Yao-Hua Ho, Hu-Cheng Lee, Hsuan-Cho Wu, Hao-Min Liu, Hsin-Hung Hsieh, Yu-Te Huang, and Shih-Chun Candice Lung. An open framework for participatory pm2. 5 monitoring in smart cities. *Ieee Access*, 5:14441–14454, 2017.
- [11] Robert Thomas Olszewski. *Generalized feature extraction for structural pattern recognition in time-series data*. Carnegie Mellon University, 2001.
- [12] Mike West, Raquel Prado, and Andrew D Krystal. Evaluation and comparison of eeg traces: Latent structure in nonstationary time series. *Journal of the American Statistical Association*, 94(446):375–387, 1999.
- [13] Pierre Geurts. Pattern extraction for time series classification. In *European conference on principles of data mining and knowledge discovery*, pages 115–127. Springer, 2001.
- [14] FK-P Chan, AW-C Fu, and Clement Yu. Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on knowledge and data engineering*, 15(3):686–705, 2003.
- [15] Eamonn J Keogh and Michael J Pazzani. Relevance feedback retrieval of time series data. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190, 1999.
- [16] Skyler Seto, Wenyu Zhang, and Yichen Zhou. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE symposium series on computational intelligence*, pages 1399–1406. IEEE, 2015.
- [17] Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2796–2802, 2013.
- [18] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR, 2017.
- [19] Ryan Flanagan and Lucas Lacasa. Irreversibility of financial time series: a graph-theoretical approach. *Physics Letters A*, 380(20):1689–1697, 2016.
- [20] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering—a decade review. *Information systems*, 53:16–38, 2015.
- [21] Rohit J Kate. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, 30:283–312, 2016.
- [22] Chotirat Ann Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 11–22. SIAM, 2004.
- [23] 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.
- [24] Zhengyu Chen and Jie Gu. High-throughput dynamic time warping accelerator for time-series classification with pipelined mixed-signal time-domain computing. *IEEE Journal of Solid-State Circuits*, 56(2):624–635, 2020.
- [25] Taha Belkhouja, Yan Yan, and Janardhan Rao Doppa. Dynamic time warping based adversarial framework for time-series domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [26] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.
- [27] Zhi Guo He and Ze Min Liu. Chinese connected word speech recognition based on derivative dynamic time warping. *Advanced materials research*, 542, 2012.
- [28] 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.
- [29] Paweł Stefaniak, Bartosz Jachnik, Wioletta Koperska, and Artur Skoczylas. Localization of lhd machines in underground conditions using imu sensors and dtw algorithm. *Applied Sciences*, 2021.
- [30] Feng Guo, Fumin Zou, Sijie Luo, Lyuchao Liao, Jinshan Wu, Xiang Yu, and Chen Zhang. The fast detection of abnormal etc data based on an improved dtw algorithm. *Electronics*, 2022.
- [31] Huanhuan Li, Jingxian Liu, Zaili Yang, Ryan Wen Liu, Kefeng Wu, and Yuan Wan. Adaptively constrained dynamic time warping for time series classification and clustering. *Information Sciences*, 534:97–116, 2020.
- [32] Ben Herbst and Hanno Coetzer. On an offline signature verification system. In *Proceedings of the 9th Annual South African Workshop on Pattern Recognition*, pages 39–43, 1998.
- [33] Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang, and Thomas Corpetti. Dynamic time warping under limited warping path length. *Information Sciences*, 393:91–107, 2017.
- [34] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.
- [35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [36] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.
- [37] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.
- [38] Jingzhao Hu, Hao Zhang, Yang Liu, Richard Sutcliffe, and Jun Feng. Bbw: a batch balance wrapper for training deep neural networks on extremely imbalanced datasets with few minority samples. *Applied Intelligence*, pages 1–16, 2022.
- [39] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.
- [40] Yuan Yuan and Lei Lin. Self-supervised pretraining of transformers for satellite image time series classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:474–487, 2020.
- [41] Peruri Venkata Anusha, Ch Anuradha, Patnala SR Chandra Murthy, and Ch Surya Kiran. Detecting outliers in high dimensional data sets using z-score methodology. *International Journal of Innovative Technology and Exploring Engineering*, 9(1):48–53, 2019.
- [42] Kaveh Samiee, Peter Kovacs, and Moncef Gabbouj. Epileptic seizure classification of eeg time-series using rational discrete short-time fourier transform. *IEEE transactions on Biomedical Engineering*, 62(2):541–552, 2014.
- [43] Alan V Oppenheim and Ronald W Schafer. 1) iscrete-” i’ime signal l’rocessing. 1989.
- [44] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.
- [45] MD Malkauthekar. Analysis of euclidean distance and manhattan distance measure in face recognition. In *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pages 503–507. IET, 2013.
- [46] BH Shekar and Guesh Dagnew. Grid search-based hyperparameter tuning and classification of microarray cancer data. In *2019 second international conference on advanced computational and communication paradigms (ICACCP)*, pages 1–8. IEEE, 2019.
- [47] Charlotte Pelletier, Silvia Valero, Jordi Inglada, Nicolas Champion, and Gérard Dedieu. Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 187:156–168, 2016.



- [48] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.



**Sachit Mahajan** is a postdoctoral researcher at the Professorship of Computational Social Science at ETH Zurich. He has PhD in Social Networks and Human-Centered Computing, and his research interests lie at the intersection of Ubiquitous Systems, Data Science, and Citizen Science.