Measuring the accuracy of time series reduction methods based on modified dynamic time warping distance calculations

Anupama Jawale[1]*, Amiya Kumar Tripathy[2]

[1]Department of Information Technology, Narsee Monjee College of Commerce and Economics, Mumbai, Maharashtra, India

[2]Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, Maharashtra, India

*Corresponding author E-mail: anupama.jawale26@gmail.com, anupama.jawale@nmcce.ac.in, amiya@dbit.in

(Received 11 September 2024; Final version received 22 September 2024; Accepted 25 November 2024)

Abstract

Representation of sensor data in the form of time series is a crucial aspect of numerous related tasks such as comparison, reduction, clustering, and classification. Time series representation methods included in most programming languages/ integrated development environments support dimensionality reduction, data preprocessing, and feature extraction for time series data, as do several normalization techniques. This research study focused on 14 different methods of dimensionality reduction from the TSepr (R Studio) package on eight different time series, which are collections of sensor data of varying lengths. The similarity of the reduced time series and the original time series is compared using a modified version of dynamic time warping with time alignment measurement. These methods are further combined with the Gaussian kernel function to normalize the distance between variously aligned series. The results showed that perceptually important points (PIP) and piecewise linear approximation (PLA) were found as the best methods for TS reduction with a minimum deviation (error term) as low as 5 – 12%. The results also indicate that PIP performs significantly differently compared to seasonal decomposition, while there are no significant differences between PIP and the other methods (PLA, FEACLIPTREND, and FEACLIP). In addition, this research study demonstrated the development of an interactive web-based application in which time series are stored in csv files, and the distance between them is calculated through the chosen reduction method.

Keywords: Dimensionality, Distance, Dynamic Time Warping, Gaussian Kernel, Time Series

1. Introduction

Many conventional waveform processing techniques can be used for a time series since it can be seen as a waveform when graphically displayed. Accelerometer data consists of three channels (x, y, and z) and is collected at a high sampling rate. This leads to a large amount of data being collected with close, continuous values against increasing unit time. For example, data collected at 100 Hz sampling data generates 6000 data points per minute per channel. Handling such large data often requires larger computational costs and storage, which makes the task challenging to process in real time. Raw accelerometer signal contains noise caused due to physical sensor inaccuracies and external vibrations. By reducing the dimensionality of this data, noise reduction and smoothening of the data make it less sensitive to noise. In this study, accelerometer data collected for road abruptions is driven in the form of time series, and dimensionality reduction techniques are presented, using 14 different methods of dimensionality reduction.

Processing accelerometer data, which consists of three channels (x, y, z) collected at high sampling rates, presents significant challenges due to the sheer volume of data generated – 6000 data points per minute per channel at 100 Hz sampling frequency. This large dataset can lead to increased computational costs and storage requirements, complicating real-time processing efforts (Hussein et al., 2024). The raw signals are often contaminated with noise from sensor inaccuracies and external vibrations, necessitating effective noise reduction techniques to enhance data quality. To address these challenges, dimensionality reduction techniques are employed, which help in reducing the data's complexity while preserving essential information (Juliusdottir, 2023). In this study, 14 distinct methods of dimensionality reduction are explored, facilitating the smoothening of data and making it less sensitive to noise. In addition, a symbolic approach is introduced to represent the data streams in a reduced space, transforming real-valued data into a string of symbols, which aids in the efficient processing of time series data. By integrating these methodologies, the study aims to improve the handling of accelerometer data collected during road abruptions, ultimately enhancing real-time analysis capabilities (Juliusdottir, 2023).

A Time series, if represented graphically, can be viewed as a waveform and hence supports many traditional methods of waveform processing. Time series are collections of data points recorded against timestamps. Sensor data are prototype examples of time series. The series under study consists of accelerometer data generated using a smartphone sensor. In general, a time series, in its simplest form, can be defined as follows in Eq. (1)

$$TS = [(dpt_1, t_1), (dpt_2, t_2), (dpt_3, t_3)\ldots (dpt_n, t_n)] \tag{1}$$

Where each *dpt* is a data point at *t* is a time at which *dpt* is measured.

Classification of time series representations has been performed by several researchers (Biemann & Masseglia, n.d.) as shown in the diagram below (Fig. 1). Non-data adaptive time representation refers to the approximation of a time series based on the local properties of the dataset. The data-adaptive representation chooses a common representation from the original time series such that while reconstructing the original time series from the reduced one, the global error is minimized. Model-based representations use a statistical model to represent the characteristics of time series (Wang et al., 2010).

The selection of the 14 dimensionality reduction methods is based on their ability to effectively manage high-dimensional time series data while preserving essential features. The implementation of distance functions, particularly the combination of dynamic time warping (DTW) and time alignment measurement (TAM), enhances the assessment of similarity between time series. Furthermore, addressing the statistical significance of results with a heat map ensures the reliability and applicability of findings, paving the way for improved analysis and interpretation of time series data across various domains. The methods used to reduce the time series considered in this research study are described in the following section.

1.1. Non-data Adaptive Methods

a. Piecewise aggregate approximation (PAA): The default algorithm of PAA uses the mean as the aggregation function. This method uses mean, max, min, sum, or any other aggregate function passed by the user. The PAA approximation is given by Eq. (2) below (Ines Silva & Henriques, 2020).

$$\bar{x}_i = \frac{M}{n}\sum_{\frac{n}{M}(i-1)+1}^{\left(\frac{n}{M}\right)i} x_i \tag{2}$$

b. Discrete wavelet transform (DWT): This function computes discrete wavelet coefficients from a given time series. The parameter *level* determines the number of coefficients, whereas the filter option provides types of wavelet filters (for example, haar, d6, and d2). The DWT divides signals into *details* and *approximate* parts. The transform contains an insignificant noise component (*details)* that can be removed or filtered out using two basic filters, thresholds, and/or wavelet types. The *Filter* parameter defines the basic waveform matching with the shape of the original waveform to be filtered out. DWT is given as follows in Eq. (3)

$$\varphi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k}\, \varphi(2x-k) \tag{3}$$

c. Discrete Fourier Transform (DFT): DFT is the primary transformation function used in digital signal processing. According to the mathematical formula, the discrete Fourier transform converts N discrete-time samples to the same number of discrete frequency samples as given by equation (4)

$$f(x) = a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L}\right) \tag{4}$$

d. Discrete Cosine Transform (DCT): DCT is the technique for converting a signal into elementary frequency components. DCT represents the input signal as a linear combination of weighted basis functions related to the frequency component. DCT can be mathematically represented as given below in Eq. (5):

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \ddot{E}(i).\cos\left[\frac{\pi.\mu}{2.N}(2i+1)\right] f(i) \tag{5}$$

e. Simple moving average (SMA): A SMA is a statistical method that calculates the mean of subsets of the dataset. The function returns a time series of length: length=length (TS)-order+1,

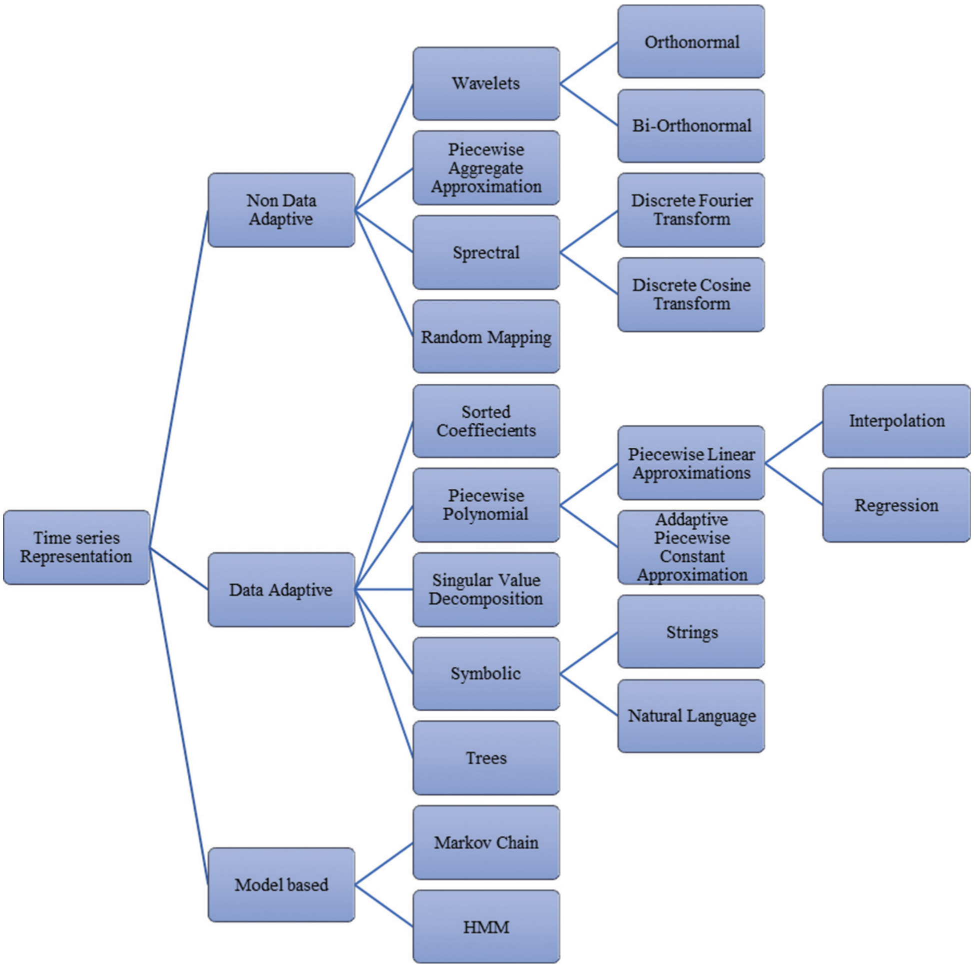

Time series Representation
Non Data Adaptive
Data Adaptive
Model based
Wavelets
Piecewise Aggregate Approximation
Sprectral
Random Mapping
Orthonormal
Bi-Orthonormal
Discrete Fourier Transform
Discrete Cosine Transform
Sorted Coeffiecients
Piecewise Polynomial
Singular Value Decomposition
Symbolic
Trees
Piecewise Linear Approximations
Addaptive Piecewise Constant Approximation
Interpolation
Regression
Strings
Natural Language
Markov Chain
HMM

Fig. 1. Classification hierarchy of time series representation

where the order is the parameter to the function, given by Eq. (6)

$$SMA = \frac{1}{M}\sum_{i=1}^{M} dtp_i \qquad (6)$$

f. Perceptually important points (PIP): PIPs are identified by extracting dominating data points from the shape of the time series (Jiménez et al., 2016). The function accepts the number of PIPs to be identified and returns them with or without the time stamp value as specified by the user.

1.2. Data Adaptive Methods

1. Symbolic aggregation approximation (SAX): This method was first proposed by Lin et al. (2000) and extends the concept of piecewise approximation. SAX is a symbolic representation of univariate time series, allowing dimensionality reduction with low storage requirements. It is applicable in motif discovery, data mining, and large-scale data processing (Camerra et al., 2010). SAX converts a time series TS of length n into a string of arbitrary length where *len(string)<<n*. To construct the alphabet, SAX uses the formula given in Eq. (7).

$$\mathbb{C}*i = alpha * j, iif \ , \overline{c} * i \in \left(\beta_{j-1} . \beta_j\right) \qquad (7)$$

Then, SAX locates the distance calculation in the lookup table of the $N \ X \ N$ matrix to construct the alphabet.

2. Piecewise linear approximation (PLA): The PLA is a method of fitting a non-linear objective function to an approximation function by adding additional variables and constraints (Lin et al., 2003). The function converts TS to a specified number of points using the PLA algorithm. The overall piecewise linear function is given by Eq. (8).

Let $x_0, x_1, x_2 \dots x_n$ where n is the number of subintervals.

Hence, each subinterval is defined as a linear function.

$$f_i(x) = m_i.(x-x_i) + b_i \qquad (8)$$

1.3. Model-Based Methods

1. Mean Seasonal Profile: This method computes the mean seasonal profile of the time series. The length of the representation can be specified by the *freq* parameter.
2. Model-based seasonal representations based on linear additive models: linear models or generalized additive models combine the properties of generalized linear models and additive models. These methods extract linear coefficients from a given time series depending upon the frequency assigned by the user. In the GAM model, Y variable depends linearly on the unknown smoothing function of certain variables. GAM is given by the following formula: Eq. (9),

$$f\left(\vec{x}\right)=\Phi\ (\sum_{p=1}^{n}\phi_p\left(x_p\right) \qquad (9)$$

3. Exponential smoothing seasonal coefficients: This function extracts exponential smoothing seasonal coefficients from the time series. This method is suitable for data that do not show any seasonal pattern or trend. Eq. (10) represents the mathematical formulation of exponential smoothing.

$$S_t = \alpha.X_t + (1-\alpha).S_{t-1} \qquad (10)$$

1.4. Data Dictated Methods

1. Feature extraction from clipped representation: This method computes features of the time series using bit-level clipped representation. It extracts 8-bit features from the data. This approach is a sustainable high-performance outlier detection method (*http://Acmbulletin.Fiit.Stuba.Sk/Vol10num2/Vol10num2.Pdf*, n.d.).
2. Feature extraction from the trending representation: similar to the clipped representation of a time series, this function extracts bit-level features but with trending. The user specifies the number of pieces and forms every piece; two features are extracted.
3. Feature extraction from clipped and trending representations: In this method, clipping and Trending both bit-level representations are combined for time series feature extraction.

To standardize the raw data series collected, normalization of time series, followed by windowing and clipping, are implicated in the proposed methodology. The normalization method of min-max normalization is used in this research study and is given in Eq. (11):

$$Y_i = \frac{X_i - \min(X)}{\max\left(X\right) - \min(X)} \qquad (11)$$

2. Related Work

Any collection of a series of data points that is observed for different points of time is a time series. The representation of sensor data as time series is essential for various tasks, including dimensionality reduction and classification. Recent studies have explored multiple methods for effectively reducing the dimensionality of time series data, highlighting their performance and applicability. Table 1 highlights some recent work in this domain.

In time series, data correlation within adjacent points in time makes time series analysis a special field of interest with special statistical features. This time correlation forms many mathematical and statistical questions with various applications in diverse fields. For example, data points collected by earthquake sensors, data points collected by temperature sensors, or brain waves in electroencephalogram. Time series patterns pose certain diverse applications of time series. In this section, we will concentrate on one special type of time series that is generated by accelerometer data signals. The accelerometer, as the data recording instrument, records vertical violation data at a frequency of 50 Hz. The general problem of interest is to classify or distinguish different types of waveforms generated by this accelerometer in case of different events observed. There are many features generated from this time series, for example, amplitude ratios, threshold of vibration, maximum amplitude, and so on. Along with these time domain features, there are several frequency domain features like spectral analysis of variance, septal coefficients, and linear prediction coefficients. In time series analysis, shape analysis is another area of interest. The shape appearance of the time series changes completely with varying sampling rates or with varying frequency and different numbers of frames of sample. Time series can differ in degrees of smoothness (Bairagi, 2018).

Moving averages and auto regression are some methods used to represent time series and used to predict time series future values. However, for recording events data with accelerometer, these techniques are not very useful as they are based on previous values in the time series (Wang et al., 2010). Autocorrelation and cross-correlation are certain methods that can be used for the comparison of similarity between two-time series. As described in the later section, electronic signal comparison techniques like DTW, SAX distances, Correlation distances, and Fourier distances are some distance measures to find out the difference between two or more time series (De Oliveira Marques et al., 2022).

Although the R programming language is a popular tool for statistical research, there are few related research papers dedicated to a specific package or functionality in R. The TSrepr basis functions are also

Table 1. Sensor data as time series and applications

Research study	Insights
Hussein et al., 2024	The authors propose a novel approach using early exit classifiers that can make accurate inferences with partial sensor data, significantly reducing energy usage while maintaining accuracy. Evaluations across six datasets demonstrate that the proposed method can achieve energy savings of 50 – 60% without compromising classification accuracy.
Meng et al., 2024	A dimension reduction method to reduce scale for time series and analyze the correlation between multi-source sensors is proposed and carried out on an industrial excavator dataset to verify the effectiveness and preponderance of the method.
He et al., 2023	In this paper, a double mean representation method, symbolic aggregate approximation based on double mean representation (SAX-DM), was proposed for time series data.
Wang et al., 2023	Wang *et al*., as discussed by the authors, proposed a multivariate time-series unsupervised domain adaptation (MTS-UDA) method to reduce the domain discrepancy at both the local and global sensor levels.
Ashraf et al., 2023	In this article, the authors present twelve different dimensionality reduction algorithms that are specifically suited for working with time-series data and fall into different categories, such as supervision, linearity, time and memory complexity, hyper-parameters, and drawbacks.

available for C++ language integration (Eddelbuettel & François, 2011) and hence open up many wide areas of research in time series. However, modern time series data with minor time intervals, such as accelerometer or sensor data, need to be studied further. Several of the time series considered by various researchers include electricity consumption and sales forecasting. Time series analysis in R programming with autoregressive integrated moving average (ARIMA) model has been employed to forecast electricity usage. Linear regression and ARIMA are used for mining time series for the women's expenditure dataset (Tanwar & Kakkar, 2017). The authors stated that the prediction accuracy is similar for both prediction models. For high-voltage load forecasting, Matsila and Bokoro (2018) used R-visualization techniques for time series and linear progression. In the study by Wang et al. (2010), all the methods are compared on the basis of similarity measures and step patterns for parameter tuning. Ali et al. (2019) review numerous methods for the visual analysis of time series data, such as clustering, classification, and other distance matrix computation methods. For a larger time series, (Camerra et al., 2010) paper defines a novel data structure called iSAX for the SAX method of aggregate approximation, which is described by the TSrepr package. The repository (Laurinec, 2018) provides the updated open-source code and documentation for the TSrepr package. The classification accuracy of all methods of the TSrepr package with aggregation and clustering methods was assessed in a previous study (Laurinec & Lucka, 2016) based on robust linear regression, exponential smoothing, and other adaptive and model methods from the package. The authors have implemented these methods for forecasting electricity consumption. Another major area of study for time series mining is in the area of motif discovery. The package TSMining (Lin et al., 2003) also implements various functions of TSrepr, but the goal is toward motif discovery from time series mining rather than dimensionality reduction of time series data.

For the multivariate time series, the alignment and similarity assessment (MTASA) framework discussed in a study by Tonle et al. (2024) integrates multiple steps of time series similarity assessment, including feature representation, alignment, and similarity measurement. With digital signal processing techniques, such as cross-correlation and convolution, MTASA enhances the alignment of time series data, addressing challenges related to noise and temporal misalignments. The implementation of a multiprocessing engine further optimizes computational resources, making the framework suitable for large-scale datasets. This method shows promise in applications such as environmental monitoring and agricultural studies, where multivariate data is prevalent. For the similarity search methods, another research study (He et al., 2023) indicates that similarity measures should not only focus on direct comparisons but also consider the underlying structures and patterns within the data. This adaptability is crucial in fields such as engineering, where degradation curves of similar systems need to be compared accurately for predictive maintenance.

On the basis of the evidence from these previous studies, we observed unexplored research on time series representation, suggesting the need for additional studies in the domain of time series reduction and distance calculation methods. Section IV of this paper explores multiple methods of distance calculation, whereas Section 5 presents the methodology of the work. The next section describes the data set used in this research study.

3. Dataset Description

The dataset used in this research study was collected by an accelerometer sensor mounted on a two-wheeler vehicle that travels on different types of roads, namely, a bituminous road, a concrete road, paved, and unpaved road (Fig. 2A-D, respectively). Smartphones' built-in triaxial accelerometer sensors are used to collect Z-axis readings, which are vertical acceleration readings against gravitational force. The roughness and surface texture of the road are reflected in accelerometer readings. The frequency of data collection was set to 50 data points per second.

A testbed was developed to aid data collection in the development and testing of this research. Fig. 3 shows the vertical placement of the smartphone with the sensors on a two-wheeler. The data collected were in the raw format, preprocessed for missing values, and normalized via the min–max normalization technique.

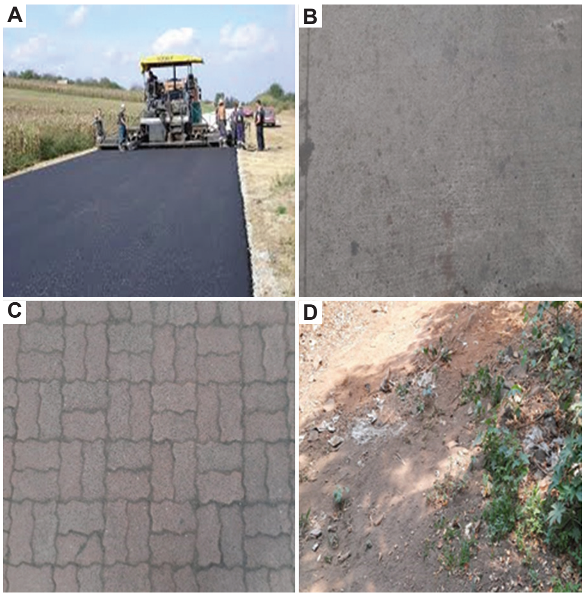


Fig. 2. (A-D) Types of roads

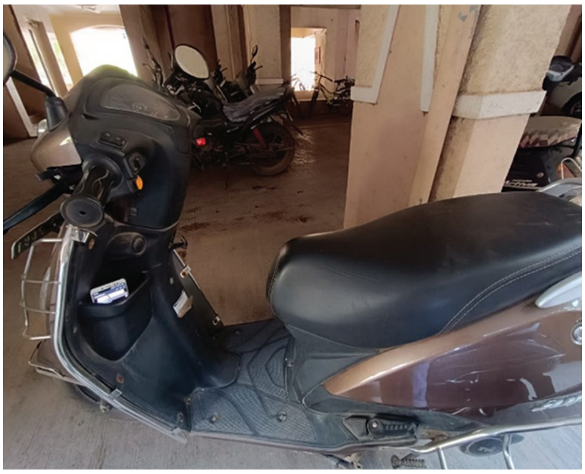


Fig. 3. Placement of smartphone

Previous studies have shown that the collection of accelerometer sensor data is not affected by the speed of the vehicle (Anand et al., 2020).

4. Distance Function

Viewing the statistical properties of time series as distance measures provides a powerful approach to understanding the behavior and relationships between time series data. Mahalanobis distance, measures of divergence, and higher-order statistics provide valuable insight into the central tendency, variability, shape, and time dependencies of time series. Careful consideration of data pre-processing, dimensionality, and complexity is essential for a meaningful application of statistical properties as distance measures. Overall, the inclusion of statistical properties in the time series analysis contributes to more accurate and robust analysis in different areas.

The distance functions used in this research study are DTW combined with TAM. Correlation-based and compression-based dissimilarity are a few more commonly used functions for time series data (Giorgino, 2009; Salvador & Chan, n.d.; Sharma et al., 2020; Singh & Meena, 2009). Eqs. 12 – 15 show the mathematical formulation of these distance functions (Montero & Vilar, 2014).

Dynamic Time Warping Distance

$$D\left(i,j\right)=\left|\mathrm{x_i}-\mathrm{y_j}\right|+min\begin{cases}D(i-1,j-2)\\D(i-1,j-1)\\D(i-2,j-1)\end{cases} \tag{12}$$

$$Time\ Alignment\ Measurement\ Distance\,\Gamma=$$
$$\vec{\varphi}+\overleftarrow{\varphi}+\left(1-\overline{\varphi}\right) \tag{13}$$

$$where\,\vec{\varphi}=\ fraction\,of\ advance\,of\ signal,\overleftarrow{\varphi}=delay$$
$$and\ \overline{\varphi}=phase$$

$$Correlaton\,based\ Distance=$$
$$1-\frac{\frac{1}{n+1}\sum_{i=1}^{n}((x_i-\overline{x}).(y_i-\overline{y}))}{\sigma X\sigma Y} \tag{14}$$

$$Compression\,based\ Dissimilarity\ Distance=$$
$$\frac{Compressed(T_1T_2)}{Compressed(T_1).Compressed(T_2)} \tag{15}$$

However, using the raw distance functions is not always guaranteed to produce the best results. The technique of collecting local neighborhood data by converting the distance to a Gaussian kernel and giving more weight to closer neighbors, can increase the distinctiveness of the similarity measure and increase the classification accuracy.

4.1. Understanding Kernel Methods

Machine learning for non-linear computing often employs a class of approaches known as kernel methods. They use the idea of feature mapping to add dimension to the original data, which could make linear processes more efficient (De Oliveira Marques et al., 2022). Kernel approaches prevent the direct calculation of the changed data points while enabling efficient calculations with a kernel function that automatically determines this feature mapping.

a. Kernel Functions:

The kernel function is the basic unit of kernel distance calculation. A kernel is a set of mathematical functions that accepts the input and converts it into the required type of output. For example, given two input vectors, a kernel function returns the inner product in the new feature space. The similarity metric between the changed data points and this inner product is identical. Typical kernel operations include:

1. Linear Kernel: The linear kernel, which is the dot product of the input vectors, represents the initial input space. The linear kernel function is defined as

$$k(x_i, x) = x_i.x \tag{16}$$

2. Polynomial Kernel: By increasing the dot product to a certain level, the polynomial kernel enables the capture of polynomial correlations between data points.

$$k\ (x_i, x) = (x_i.x)^e \tag{17}$$

3. Gaussian – Radial Basis Function Kernel: The Gaussian kernel uses the radial basis function to calculate similarity. It gives closer locations more similarity and decreases with distance.

$$k\left(x_i, x\right) = e^{-\|x_i - x\|^2 / 2\sigma^2} \tag{18}$$

4. Sigmoid Kernel: The sigmoid kernel models sigmoidal interactions between data points by capturing similarity based on the hyperbolic tangent function.

$$k(x_i, x) = tanh\ (cx_i\, x + h) \tag{19}$$

b. Calculation of the Kernelized Distance:

We use the kernel trick concept to determine the kernel distance between two data series. We can directly compute the kernel function values between the input vectors instead of explicitly computing the feature vectors and computing distances in transformed space. Without explicitly computing the changed vectors, the kernel distance captures the disparity between data points in the changed feature space.

In this research study, to calculate kernel distance, we use the equation of Gaussian kernel as illustrated in Eq. 18.

5. Methodology

This research study focused on the accuracy of the dimensionality reduction techniques of the TSrepr package. 8 different length time series presenting Z – Acceleration of smartphone accelerometer data during different vehicle travels are considered for the analysis. These data are normalized to bring all the data points to the same scale and range of values. The various dimensionality reduction methods listed in Section 1 are implemented on this dataset. The resulting reduced dataset series is compared with the original series for similarity using the normalized distance of the DTW+TAM method. The distance between two series is given by the normalized cumulative distance. This method is highly adaptable and can be applied to a wide range of domains, including finance, healthcare, and environmental science.

The combination of DTW and TAM allows a comprehensive measure of distance calculation that measures both features of time series, temporal alignment, and magnitude difference. Eq. 20 shows the mathematical formulation of combining DTW and TAM.

$$Distance(x,y) = \alpha.DTW\ (x,y) + (1\text{-}\alpha).TAM\ (x,y) \tag{20}$$

α is a weight parameter that can be adjusted according to the weightage to be assigned to the respective factor. To ensure appropriate scaling of the distance metric, a normalization followed by Gaussian kernel-based distance calculation is applied Eq. (21).

$$k\left(x_i, x_j\right) = 1 - e^{\frac{-\omega^2}{2\sigma^2}} \tag{21}$$

ω represents the normalized distance of two-time series as given in Eq. 20; σ is a parameter that controls the width or scale of the Gaussian kernel.

The process of combining and normalizing data has the potential to enhance the accuracy of similarity assessments, thereby improving the performance of applications such as clustering, classification, and anomaly detection. Fig. 4 shows the sequence of steps of this methodology.

When the two series are the same, the normalized cumulative distance between them is zero. Any deviation from the value of zero is considered an error term. In Section 5 of this paper, we have presented the difference between the original series and the reduced series as a result of our experimental work.

In addition, the flexible integration of similarity measurements into various algorithms using the

Gaussian kernel improves their performance and enables more effective data analysis and decision-making. This research study attempts to solve this problem by following a two-step process, namely, calculating the distance between two-time series and applying a kernel function to map this distance onto a separable plane. Step 1 includes calculation of the DTW and TAM in normalized form, and Step 2 implements the Gaussian kernel function, as discussed in the above section.

Furthermore, this study presents an interactive, integrated application to improve the usability of distance calculations for exploring and analyzing time series data. Users can input time series datasets, preprocess the data, carry out reduction activities, and interactively visualize the outcomes using the program. The application will make it simple for users to extract useful insights from their time series data by offering an intuitive user interface. The algorithm for creating the application is given below.

Algorithm: Application

Import the necessary libraries: shiny and TSdist.

```
the UI:
Create a fluid page.
  Define the server
Extract the data from the uploaded CSV files.
Convert the data into numeric vectors.
Define Time series Reduction Methods
reduced_ts <- reduced_timeseries (methodname (), original_ts)
Diff <-DTW+TAM (original_ts, reduced_ts)
Diff -> 0 indicates effective reduction without loss
Redirect output to server
```

Algorithm: Distance Calculation

```
Define the Gaussian kernel function.
Input: x (input value)
Output: Gaussian kernel value using the input value and a fixed sigma value
$\delta = Dynamic Time Warping\ Distance\ D(i,j) = |x_i - y_j| + min \begin{cases} D(i-1, j-2) \\ D(i-1, j-1) \\ D(i-2, j-1) \end{cases}$
$Time\ Alignment\ Measurement\ Distance\ \Gamma = \vec{\varphi} + \overleftarrow{\varphi} + (1 - \overline{\varphi})$
$where \vec{\varphi} = fraction of\ advance of\ signal, \overleftarrow{\varphi} = delay and\ \overline{\varphi} = phase$
$combined\ distance \mathsf{C} = \alpha.\delta + (1-\alpha).\Gamma$
$Normalized\ distance\ \omega = \frac{\mathsf{C}}{Max\_\partial}$
$Difference \Delta = 1 - e^{\frac{-\omega^2}{2\sigma^2}}$
Output: return (Δ)
```

6. Results and Discussion

The combination of alignment cost (DTW) and magnitude difference (TAM) provides a comprehensive measure of similarity, taking into account both temporal alignment and differences in values. The measure can be adjusted according to specific needs using a weighted average of distances. The normalization of the Gaussian kernel ensures that the distance metric is appropriately scaled, facilitating its interpretation and comparison across diverse datasets.

The Gaussian Kernal makes DTW distance in the form of a positive semi-definite (PSD) matrix, a symmetric matrix with non-negative eigenvalues. The PSD matrix is defined as

$$M \epsilon L(V), where\ M\ is\ symmetric\ and\ v^T M_v > 0\ \forall\ v \in V \quad (22)$$

The PSD matrix ensures that the SVM algorithm will terminate at a global optimum, which leads to a more interpretable and reliable solution.

In addition, it improves the detection of similarities by identifying similarities that may go unnoticed when relying solely on a single measure.

Table 2 shows the tabulated results of all the normalized cumulative distances between the reduced series and the original series. This is the error term given by |0-NormDist|. Table 3 shows the five methods with the minimum error term calculated using the formula given in Eq. (23).

$$Percentage\ Error\ Term = |0\text{-}NormDist| * 100 \quad (23)$$

The PIP, PLA, seasonal decomposition (SEAS), feature extraction and clipping for trend (FEACLIPTREND), and feature extraction and clipping (FEACLIP) methods are advanced strategies for dimensionality reduction in time series analysis. Let X ∈ R^{nXm} represents a time series with n observations and m features. The goal of dimensionality reduction is to transform *X* into a lower dimension representation $Y \in R^{nXk}$ where $k<m$. The understudied methods aim to retain certain statistical features of a time series data with reduced dimensionality, as described below.

Fig. 5 shows the results of the heatmap visualization. Fig. 6 shows a visualization of the original time series and reduced time series. The PIP, PLA, SEAS, FEACLIPTREND, and FEACLIP methods yield the best results by considerably reducing the dimensionality but keeping the original features intact since the distance between the original and reduced time series is significantly

PIP (Principal Information Preservation	**Y=XW, where W is projection matrix such that** $\text{maximize}\operatorname{Var}(Y), W_F^2=1$
PLA (Piecewise Linear Approximation)	$Y_i = \sum_{j=1}^{k} a_j . 1_{t_j t_{j+1}}(t_i)$ where a_j is coefficient of linear segment, 1 is an indicator function to check if t_i falls within a range
SEAS (Seasonal Decomposition)	*X (t) = T (t) + S (t) + R (t)* where T: Trend, S: Seasonality and R: Residual
FEACLIPTREND (Feature Extraction and Clipping for Trend)	*Y=Clip (X, ϵ)* where ϵ is threshold to determine the significant feature to be retained
FEACLIP (Feature Extraction and Clipping	*Y=Extract (X)∩Clip (X,ϵ)*

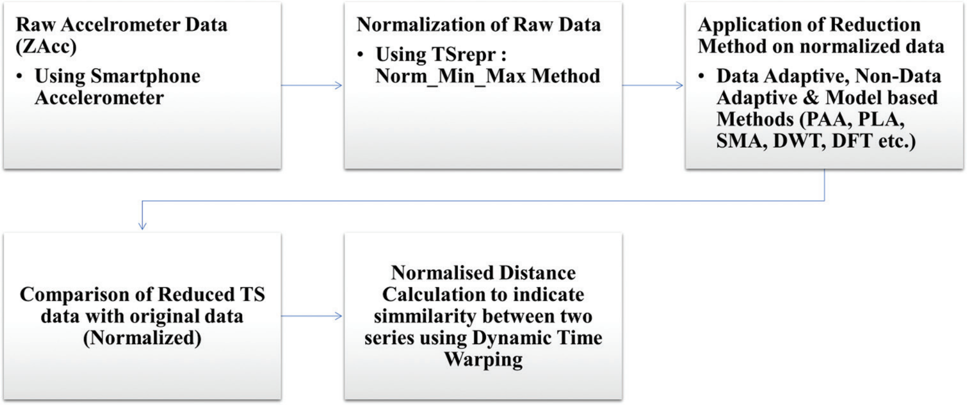
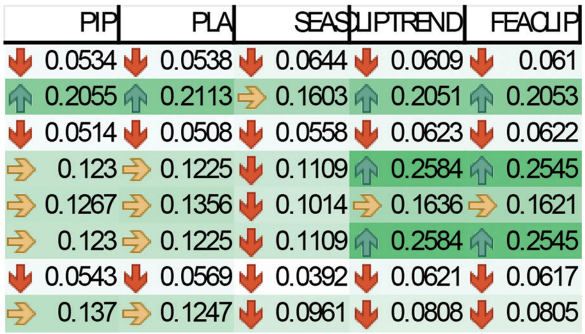

Raw Accelrometer Data (ZAcc)
• Using Smartphone Accelerometer
Normalization of Raw Data
• Using TSrepr : Norm_Min_Max Method
Application of Reduction Method on normalized data
• Data Adaptive, Non-Data Adaptive & Model based Methods (PAA, PLA, SMA, DWT, DFT etc.)
Comparison of Reduced TS data with original data (Normalized)
Normalised Distance Calculation to indicate simmilarity between two series using Dynamic Time Warping

Fig. 4. Methodology flow



PIP	PLA	SEAS	LIPTREND	FEACLIP
↓ 0.0534	↓ 0.0538	↓ 0.0644	↓ 0.0609	↓ 0.061
↑ 0.2055	↑ 0.2113	→ 0.1603	↑ 0.2051	↑ 0.2053
↓ 0.0514	↓ 0.0508	↓ 0.0558	↓ 0.0623	↓ 0.0622
→ 0.123	→ 0.1225	↓ 0.1109	↑ 0.2584	↑ 0.2545
→ 0.1267	→ 0.1356	↓ 0.1014	→ 0.1636	→ 0.1621
→ 0.123	→ 0.1225	↓ 0.1109	↑ 0.2584	↑ 0.2545
↓ 0.0543	↓ 0.0569	↓ 0.0392	↓ 0.0621	↓ 0.0617
→ 0.137	→ 0.1247	↓ 0.0961	↓ 0.0808	↓ 0.0805

Fig. 5. Heat map representation of the distance between original and reduced time series

lower (ow – 20%), as shown in Table 3. The dimensionality reduction procedure is presented in Table 4.

We conducted a paired t-test to verify the results of the differences of the top five methods. The results obtained are listed as

i. PIP versus PLA: Statistic: −0.216, p-value: 0.835: No significant difference between PIP and PLA.
ii. PIP versus SEAS: Statistic: 2.423, p-value: 0.046: There is a statistically significant difference between PIP and SEAS at the 0.05 significance level.
iii. PIP versus FEACLIPTREND: Statistic: −1.454, p-value: 0.189: No significant difference between PIP and FEACLIPTREND.
iv. PIP versus FEACLIP: Statistic: −1.437, p-value: 0.194: No significant difference between PIP and FEACLIP.

The results indicate that PIP performs significantly differently compared to SEAS, while there are no significant differences between PIP and the other methods (PLA, FEACLIPTREND, and FEACLIP).

Fig. 5 shows a heat map visualization of the percentage error term. The color scale of light yellow to green shows minimum error terms to maximum error terms. Fig. 6 shows a visualization of the time series original, PAA, SMA, and PIP.

The dimensionality reduction methods exhibit substantial potential for real-world applications and integration into existing systems. These techniques effectively reduce the dimensionality of time series data while preserving the integrity of original features, which is crucial for enhancing interpretability, efficiency, and performance across various domains.

Fig. 7 shows the interactive web application used to compare two-time series with four different distance measures.

Table 2. Experimental results

a. Non-data adaptive methods

Normalized cumulative distance between original and reduced time series

Time series	LEN	SMA	RDWT	DFT	DCT	PAA	PIP
TS1	101	0.0538	0.06276	0.62946	0.15076	0.05339	0.06438
TS2	101	0.21127	0.26822	0.89585	0.43408	0.20551	0.16026
TS3	101	0.05082	0.05375	0.46697	0.12836	0.05137	0.05579
TS4	843	0.12247	0.4158	1	0.993	0.12296	0.1109
TS5	1764	0.13563	0.32768	0.99997	0.67835	0.12666	0.10136
TS6	843	0.12247	0.4158	1	0.993	0.12296	0.1109
TS7	526	0.05692	0.94684	1	1	0.05432	0.03923
TS8	2000	0.12471	0.19542	0.99858	0.67039	0.13698	0.09614

b. Data adaptive methods

Normalized cumulative distance between original and reduced time series

Time series	LEN	PLA
TS1	101	0.042641
TS2	101	0.213402
TS3	101	0.039116
TS4	843	0.416696
TS5	1764	0.527422
TS6	843	0.416696
TS7	526	0.117633
TS8	2000	0.198984

c. Model-based methods

Normalized cumulative distance between original and reduced time series

Time series	LEN	SEAS	LM	GAM	EXP
TS1	101	0.05153	0.051345	0.259189	0.179786
TS2	101	0.436253	0.436253	0.996679	0.443483
TS3	101	0.053389	0.053389	0.228889	0.196029
TS4	843	0.563937	0.563878	1	0.441416
TS5	1764	0.53348	0.533461	1	0.144275
TS6	843	0.563937	0.563878	1	0.441416
TS7	526	0.059751	0.05981	0.998337	0.299404
TS8	2000	0.16548	0.16548	0.99447	0.314765

d. Data-dictated methods

Normalized cumulative distance between original and reduced time series

Time series	LEN	FEATREND	FEACLIP	FEACLIPTREND
TS1	101	0.060922	0.060967	0.563687
TS2	101	0.205051	0.205281	0.999998
TS3	101	0.06234	0.062194	0.408944
TS4	843	0.258353	0.254489	1
TS5	1764	0.163569	0.162071	1
TS6	843	0.258353	0.254489	1
TS7	526	0.062134	0.061691	0.999569
TS8	2000	0.080842	0.08054	1

Table 3. Five best methods with the percentage of error term

% Error between original and reduced time series					
TS-LEN	**PIP (%)**	**PLA (%)**	**SEAS (%)**	**FEACLIPTREND (%)**	**FEACLIP (%)**
TS1-101	5.34	5.38	6.44	6.09	6.10
TS2-101	20.55	21.13	16.03	20.51	20.53
TS3-101	5.14	5.08	5.58	6.23	6.22
TS4-843	12.30	12.25	11.09	25.84	25.45
TS5-1764	12.67	13.56	10.14	16.36	16.21
TS6-843	12.30	12.25	11.09	25.84	25.45
TS7-526	5.43	5.69	3.92	6.21	6.17
TS8-2000	13.70	12.47	9.61	8.08	8.05

Table 4. Reduction percentage in size

TS-LEN	PIP (%)	PLA (%)	SEAS (%)	FEACLIPTREND (%)	FEACLIP (%)
TS1-101	89.11	89.11	90.10	88.12	92.08
TS2-101	89.11	89.11	90.10	88.12	92.08
TS3-101	89.11	89.11	90.10	88.12	92.08
TS4-843	15.84	89.11	16.83	88.12	92.08
TS5-1764	79.00	98.70	79.12	98.58	99.05
TS6-843	95.18	99.38	95.24	99.32	99.55
TS7-526	93.59	98.70	93.71	98.58	99.05
TS8-2000	61.79	97.91	61.98	97.72	98.48

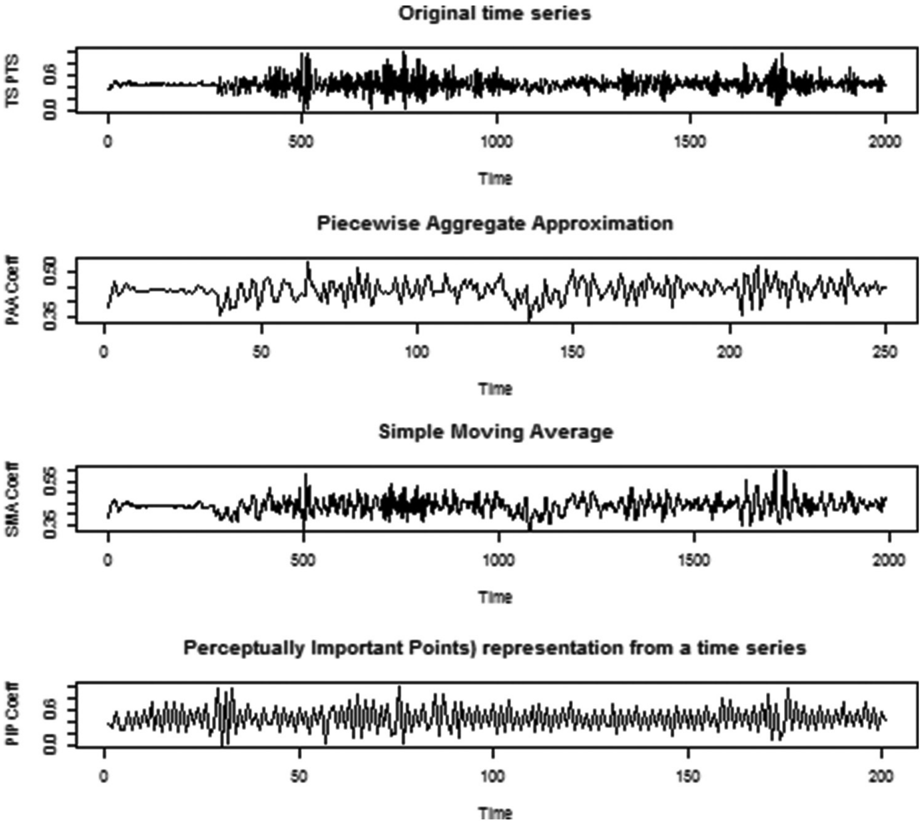
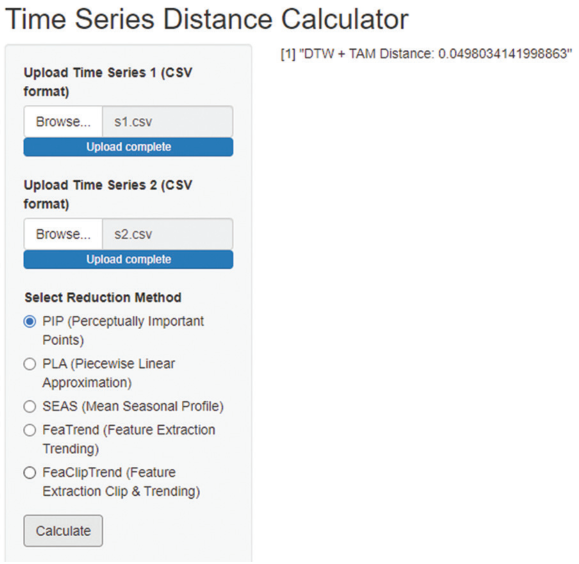


Fig. 6. Piecewise aggregate approximation, simple moving average, and perceptually important points visualization with original time series



Time Series Distance Calculator

[1] "DTW + TAM Distance: 0.0498034141998863"

Upload Time Series 1 (CSV format)

Browse... s1.csv

Upload complete

Upload Time Series 2 (CSV format)

Browse... s2.csv

Upload complete

Select Reduction Method

● PIP (Perceptually Important Points)
○ PLA (Piecewise Linear Approximation)
○ SEAS (Mean Seasonal Profile)
○ FeaTrend (Feature Extraction Trending)
○ FeaClipTrend (Feature Extraction Clip & Trending)

Calculate

Fig. 7. Web Application for distance calculation

The purpose of this Shiny app, titled "Time Series Distance Calculator," is to assist users in uploading two-time series datasets, implementing a selected reduction method, and computing the similarity distance between them using a combination of DTW and Time Asynchronous Matching (TAM) distances, and presenting the results in a visual format.

The application design uses *Shiney* framework in R programming language, combined with a reactive programming model for real-time updates. The application can be accessed through Shiney Server or a browser.

This application is beneficial for individuals engaged in the analysis of time series data who require a means of assessing the resemblance between two series. This interface is valuable in domains where time series analysis holds significance. Some of the use cases for the potential use of this application are Market Data Comparison, Portfolio Management, Disease Progression Monitoring Economic Indicators, and Social Trends Analysis.

7. Conclusion and Future Scope

This research study considered 14 different methods of dimensionality reduction for time series from the TSrepr package in the "R" programming. The basis for comparison is the similarity of the reduced time series with the original time series. The results revealed that the PIP and PLA methods reduce the dimensionality of the time series by 90 – 95%. Furthermore, by comparing these time series on the basis of the combined warping path and magnitude, a novel method of time series similarity search is presented. In the future, we wish to explore other methods of the TSrepr package for dimensionality reduction of multivariate time series.

8. Limitations

This study primarily focuses on univariate time series, which limits the generalizability of the findings to multivariate time series data, a common scenario in real-world applications. Furthermore, the basis for comparison is the similarity between the reduced and original time series, which may not account for other important aspects, such as preserving specific patterns or trends relevant to multiple domains.

References

Ali, M., Alqahtani, A., Jones, M.W., & Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7, 181314–181338. https://doi.org/10.1109/ACCESS.2019.2958551

Anand, A., Gawande, R., Jadhav, P., Shahapurkar, R., Devi, A., & Kumar, N. (2020). Intelligent vehicle speed controlling and pothole detection system. *E3S Web of Conferences*, 170, 02010. https://doi.org/10.1051/e3sconf/202017002010

Ashraf, M., Anowar, F., Setu, J.H., Chowdhury, A.I., Ahmed, E., Islam, A., & Al-Mamun, A. (2023). A survey on dimensionality reduction techniques for time-series data. *IEEE Access*, 11, 42909–42923. https://doi.org/10.1109/ACCESS.2023.3269693

Bairagi, V. (2018). EEG signal analysis for early diagnosis of Alzheimer disease using spectral and wavelet based features. *International Journal of Information Technology*, 10(3),403–412. https://doi.org/10.1007/s41870-018-0165-5

Biemann, D.C., & Masseglia, F. (n.d.). *Time Series Clustering in the Field of Agronomy Cluster Analyse Agronomischer Zeitreihen*. Master-Thesis, p70.

Camerra, A., Palpanas, T., Shieh, J., & Keogh, E. (2010). iSAX 2.0: Indexing and Mining One Billion Time Series. In: *2010 IEEE International Conference on Data Mining*, p58–67. https://doi.org/10.1109/ICDM.2010.124

De Oliveira Marques, E.S., Alves, K.S.T.R., Pekaslan, D., & De Aguiar, E.P. (2022). Kernel Evolving Participatory Fuzzy Modeling for Time Series Forecasting: New Perspectives Based on Distance Measures. In: *2022 IEEE International Conference on Fuzzy Systems* (*FUZZ-IEEE*), p1–8.

https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882602

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40(8), 1–18. https://doi.org/10.18637/jss.v040.i08

Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7), 1–24. https://doi.org/10.18637/jss.v031.i07

He, Z., Zhang, C., & Cheng, Y. (2023). Similarity measurement and classification of temporal data based on double mean representation. *Algorithms*, 16(7), 347. https://doi.org/10.3390/a16070347

(n.d.). Available from: https://acmbulletin.fiit.stuba.sk/vol10num2/vol10num2.pdf

Hussein, D., Nelson, L., & Bhat, G. (2024). Sensor-aware classifiers for energy-efficient time series applications on IoT devices (arXiv:2407.08715). arXiv. https://doi.org/10.48550/arXiv.2407.08715

Ines Silva, M., & Henriques, R. (2020). Exploring Time-series Motifs through DTW-SOM. In: *2020 International Joint Conference on Neural Networks* (*IJCNN*), p1–8. https://doi.org/10.1109/IJCNN48605.2020.9207614

Jiménez, P., Nogal, M., Caulfield, B., & Pilla, F. (2016). Perceptually important points of mobility patterns to characterise bike sharing systems: The Dublin case. *Journal of Transport Geography*, 54, 228–239. https://doi.org/10.1016/j.jtrangeo.2016.06.010

Juliusdottir, T. (2023). topr: An R package for viewing and annotating genetic association results. https://doi.org/10.21203/rs.3.rs-2499681/v1

Laurinec, P. (2018). TSrepr R package: Time series representations. *Journal of Open Source Software*, 3(23), 577. https://doi.org/10.21105/joss.00577

Laurinec, P., & Lucka, M. (2016). Comparison of Representations of Time Series for Clustering Smart Meter Data. In: *Proceedings of the World Congress on Engineering and Computer Science* (*WCECS 2016*), p6.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (*DMKD '03*), p2. https://doi.org/10.1145/882082.882086

Matsila, H., & Bokoro, P. (2018). Load Forecasting Using Statistical Time Series Model in a Medium Voltage Distribution Network. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, p4974–4979. https://doi.org/10.1109/IECON.2018.8592891

Meng, J., Huo, X., He, C., & Zhu, C. (2024). Dimension Reduction of Multi-Source Time Series Sensor Data for Industrial Process. In: *2024 IEEE 33rd International Symposium on Industrial Electronics* (*ISIE*), p1–6. https://doi.org/10.1109/ISIE54533.2024.10595725

Montero, P., & Vilar, J.A. (2014). TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1), 1–43. https://doi.org/10.18637/jss.v062.i01

Ngabesong, R., & McLauchlan, L. (2019). Implementing "R" Programming for Time Series Analysis and Forecasting of Electricity Demand for Texas, USA. In: *2019 IEEE Green Technologies Conference* (*GreenTech*), p1–4. https://doi.org/10.1109/GreenTech.2019.8767131

Salvador, S., & Chan, P. (n.d.). FastDTW: Toward accurate dynamic time warping in linear time and space.

Sharma, S.K., Phan, H., & Lee, J. (2020). An application study on road surface monitoring using DTW based image processing and ultrasonic sensors. *Applied Sciences*, 10(13), 4490. https://doi.org/10.3390/app10134490

Singh, V., & Meena, N. (2009). Engine Fault Diagnosis using DTW, MFCC and FFT. In: U. S. Tiwary, T. J. Siddiqui, M. Radhakrishna, & M. D. Tiwari (Eds.), *Proceedings of the First International Conference on Intelligent Human Computer Interaction*. Springer, India, p83–94. https://doi.org/10.1007/978-81-8489-203-1_6

Tanwar, H., & Kakkar, M. (2017). Performance Comparison and Future Estimation of Time Series Data Using Predictive Data Mining Techniques. In: *2017 International Conference on Data Management, Analytics and Innovation* (*ICDMAI*), p9–12. https://doi.org/10.1109/ICDMAI.2017.8073477

Tonle, F., Tonnang, H., Ndadji, M., Tchendji, M., Nzeukou, A., Senagi, K., & Niassy, S. (2024). Advancing multivariate time series similarity assessment: An integrated computational approach (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2403.11044

Wang, X., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2010). Experimental comparison of representation methods and distance measures for time series data. arXiv:1012.2789 [Cs]. https://doi.org/10.48550/arXiv.1012.2789

Wang, Y., Xu, Y., Yang, J., Chen, Z., Wu, M., Li, X., & Xie, L. (2023). SEnsor alignment for multivariate time-series unsupervised domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8), 10253–10261. https://doi.org/10.1609/aaai.v37i8.26221

AUTHOR BIOGRAPHIES

Anupama Jawale received a B.Sc. (Computer Science) and MCM (Master of Computer Management) degree from North Maharashtra University, Maharashtra, India. She received an MCA (Master of Computer Applications) from Sikkim Manipal University, Sikkim, India. She received an M.Phil. (Information Technology) degree from YCMO University, Maharashtra, India. She received PhD degree (Computer Science) from SNDT Women's University, Mumbai, India in 2024. She began her academic career in affiliated institutions of the University of Mumbai, India, as a lecturer teaching undergraduate and graduate courses in computer science and Information Technology (IT). Later in 2013, she joined as an assistant professor in the department of IT at NM College of Commerce & Economics, Mumbai, India, and currently, she is heading the IT Department. Her work aims to improve knowledge and usage of data-driven methods in computer vision, time series analysis, and human-computer interaction psychology. She has contributed to addressing complex issues in a variety of industries, such as digital security, automotive, and finance, by utilizing cutting-edge approaches and optimization techniques. She is a co-author of about a dozen papers published in journals/conferences. Her current research interests are in Data Science, Feature Extraction for Sensor Data, Sensor-driven automation, and Time Series Analysis.

Amiya Kumar Tripathy is currently a Professor in the Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, India, affiliated with the University of Mumbai. He earned a PhD degree (2013) in Computer Science & Engineering (in the domain of Data Mining & Wireless Sensor Networks) from the Indian Institute of Technology Bombay, Mumbai, India. He was an adjunct associate professor in the faculty of Science & Engineering at Edith Cowan University (ECU), Australia (2014 – 2017) and later an adjunct professor in the School of Science, ECU, Australia (2017 – 2023). He had been a visiting researcher at the Rajamangala University of Technology, Bangkok, Thailand, for IoT-enabled remote monitoring of the Precision Agriculture Farming project (2017 – 2018). He has been in the software industry, research, and academia for more than two decades, having around 150 publications in journals/conference papers. His research focuses on data science, computer vision, remote sensing, and IoT for Precision Agriculture. He has contributed to numerous collaborative research and consultancy projects in the domain of data analytics in India and abroad. He has served on the technical program committees of several international conferences, has been invited as a plenary speaker, and has co-chaired sessions at various conferences.