MethodsX 13 (2024) 102840

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

asMODiTS: An application of surrogate models to optimize Time Series symbolic discretization through archive-based training set update strategy ☆

Aldo Márquez-Grajales [a], Efrén Mezura-Montes [a], Héctor-Gabriel Acosta-Mesa [a], Fernando Salas-Martínez [b,*]

[a] *Artificial Intelligence Research Institute, University of Veracruz, Campus Sur Paseo Lote II, Sección Segunda N° 112, Nuevo Xalapa, Xalapa, Veracruz 91097, Mexico*

[b] *El Colegio de Veracruz, Carrillo Puerto 26, Zona Centro, Xalapa, Veracruz 91000, Mexico*

ARTICLE INFO

Method name:
Archive-based and surrogate-assisted multi-objective approach for symbolic discretization of time series (asMODiTS)

Keywords:
Time series
Symbolic discretization
Surrogate models
Archive-based update strategy
Emodits

ABSTRACT

The enhanced multi-objective symbolic discretization for time series (*eMODiTS*) uses an evolutionary process to identify the appropriate discretization scheme in the Time Series Classification (TSC) task. It discretizes using a unique alphabet cut for each word segment. However, this kind of scheme has a higher computational cost. Therefore, this study implemented surrogate models to minimize this cost. The general procedure is summarized below.

- The K-nearest neighbor for regression, the support vector regression model, and the Ra- dial Basis Functions neural networks were implemented as surrogate models to estimate the objective values of *eMODiTS*, including the discretization process.
- An archive-based update strategy was introduced to maintain diversity in the training set.
- Finally, the model update process uses a hybrid (fixed and dynamic) approach for the surrogate model's evolution control.

Specifications table

Subject area:	Computer Science
More specific subject area:	Symbolic discretization of time series
Name of your method:	Archive-based and surrogate-assisted multi-objective approach for symbolic discretization of time series (asMODiTS)
Name and reference of original method:	Márquez-Grajales, A., Mezura-Montes, E., Acosta-Mesa, HG., Salas-Martínez, F. (2024). Use of a Surrogate Model for Symbolic Discretization of Temporal Data Sets Through eMODiTS and a Training Set with Varying-Sized Instances. In: Calvo, H., Martínez-Villaseñor, L., Ponce, H., Zatarain Cabada, R., Montes Rivera, M., Mezura-Montes, E. (eds) Advances in Computational Intelligence. MICAI 2023 International Workshops. MICAI 2023. Lecture Notes in Computer Science, vol 14,502. Springer, Cham. 10.1007/978–3–031–51940–6_27

(*continued on next page*)

☆ **Related research article:** Marquez-Grajales, A., Acosta-Mesa, H. G., Mezura-Montes, E., Hernandez-Jimenez, R., Perez-Castro, N., Aguilar-Justo, A. E., & Salas-Martinez, F. (2021). Classification of colposcopic images using a multi-breakpoints discretization approach on temporal patterns. *Biomedical Signal Processing and Control*, *69*, 102918, DOI: https://doi.org/10.1016/j.bspc.2021.102918.
DOI of original article: 10.1016/j.bspc.2021.102918

* Corresponding author.
E-mail address: fernando.salas@colver.info (F. Salas-Martínez).

https://doi.org/10.1016/j.mex.2024.102840
Received 21 March 2024; Accepted 2 July 2024
Available online 3 July 2024

	Márquez-Grajales, A.; Mezura-Montes, E.; Acosta-Mesa, H.G.; Salas-Martínez, F. A Surrogate-Assisted Symbolic Time-Series Discretization Using Multi-breakpoints and a Multi-objective Evolutionary Algorithm. Preprints 2024, 2,024,061,126. 10.20944/preprints202406.1126.v1
Resource availability:	https://www.timeseriesclassification.com/dataset.php https://github.com/scoramg/asMODiTS

Background

This paper presents a method to minimize the computational cost of the symbolic discretization process performed by the SAX-based method called eMODiTS [1]. The method aims to find discretization schemes where each slice on the time axis contains its own set of slices in the value space of the time series. Due to this characteristic, eMODiTS consumes significant computational resources, as it has an ample search space. Therefore, the proposed method aims to use surrogate models to estimate the objective functions that guide the search in eMODiTS, thereby reducing its computational cost. The discretization process required before evaluating each scheme is the cause of the high computational cost. An external file was also implemented to store the functions with the highest estimation error to provide greater predictive power to the surrogate models. This feature makes it different from those investigations where surrogate models have been implemented in the discretization or classification of time series, which, to our knowledge, are scarce.

Márquez-Grajales et al. [2] introduced the first surrogate model that handled training sets with different-sized instances. The study tested the k Nearest Neighbor algorithm (KNN) for a regression task, using values of $k = 1, 3, 5, 7, 9$ and the Dynamic Time Warping (DTW) distance instead of the Euclidean Distance. The results suggested that values of $k = 1, 9$ yielded the most competitive results compared to other versions. However, the value of $k = 9$ achieved the highest number of best accuracies in the datasets tested. The primary drawback of this approach is the surrogate model update mechanism, which utilized the solutions of the best Pareto front for this purpose, resulting in a loss of diversity in the training set and reduced accuracy predictions. Moreover, an extension of this work is presented in [3], where a modification of the surrogate model update process is proposed. These research works are the baseline of our method.

Method details

Surrogate-assisted evolutionary algorithms (SAEA)

Efficient computational models are used in surrogate-assisted evolutionary computation to approximate the fitness function in evolutionary algorithms [4]. These models are inserted into the evolutionary algorithm to evaluate new individuals and minimize the number of evaluations in the original functions. SAEA stages are described as follows [5].

- *Offline training stage*. A set of individuals evaluated in the original objective functions is created in this stage. Typically, this set is called the training set because the surrogate model uses it to estimate the values of the original functions. This process is performed only once when the evolutionary process begins.
- *Online training stage*. This stage is carried out during each generation of the evolutionary process. Here, the individuals evaluated in the original functions are used to update the training set and the surrogate model.
- *Evolution control*. Executing the previous stage requires choosing an appropriate update strategy to avoid poorly trained surrogate models. Evolution control or model management is responsible for this task. Two categories for this process are emphasized: fixed and dynamic evolution control [6]. In fixed evolution control, surrogate models can be applied at a specific moment (generation-based strategy), in a specific number of individuals (individual- based strategy), or applied in a subset of population co-evolve using a different surrogate model (population-based strategy). On the contrary, dynamic evolution control is executed regarding the accuracy of the surrogates.

Surrogate models

K-nearest neighbor regressor (KNN)

One of the most straightforward approaches for time series regression is the k-nearest neighbor (KNN) method. This distance-based method searches for k instances closer to the instance to predict. Then, a counting vote of the labels found in these k instances is performed. In classification, the label with the highest frequency is assigned to the instance of interest. On the other hand, the mean value of the labels k closer to the interest instance is assigned in the regression task [7]. Eq. (1) expresses the KNN for regression process.

$$y_{new} = \frac{1}{K}\sum_{i=1}^{K} y^{(i)}$$

where $y^{(i)}$ is obtained as follows :

$$dist_{min}(A) = min\ dist\left(x_{new}, a\right), a \in A$$

$$\left(x^{(1)}, y^{(1)}\right) = \{(x, y) \in X_{train} : dist\left(x_{new}, x\right) = dist_{min}\left(X_{train}\right)\}$$

$$
\begin{aligned}
X_{train} &= X_{train} \setminus \left(x^{(1)}, y^{(1)}\right) \\
\left(x^{(2)}, y^{(2)}\right) &= \{(x, y) \in X_{train} : dist\left(x_{new}, x\right) = dist_{min}\left(X_{train}\right)\} \\
X_{train} &= X_{train} \setminus \left(x^{(2)}, y^{(2)}\right) \\
&\vdots \\
\left(x^{(K)}, y^{(K)}\right) &= \{(x, y) \in X_{train} : dist\left(x_{new}, x\right) = dist_{min}\left(X_{train}\right)\}
\end{aligned} \tag{1}
$$

where X_{train} is the training set, Y_{train} is the training label set, x_{new} is the prediction instance, y_{new} is the predicted value, $dist(\cdot)$ is a distance function, and $dist_{min}(A)$ is the minimum distance of all the points of a set A with respect to the rest.

Radial basis functions neural networks (RBFN)

A radial basis function neural network is an artificial neural network composed of three layers: an input layer, a hidden layer that contains radial basis functions (RBF) as a transfer function, and a linear output layer [8,9].

In particular, an RBF is a mathematical function in which the distance between the measured and sampled vectors is used as the independent variable. Given n distinct sample points, the RBF can be represented as shown in Eq. (2), where λ represents the set of weights and $\phi(\cdot)$ denotes the basis function [10]. A pool of basis functions can be chosen, such as linear ($\phi(dist) = dist$), cubic ($\phi(dist) = dist^3$), thin plate spline ($\phi(dist) = dist^2 log(dist)$), multi-square ($\phi(dist) = \sqrt{dist^2 + \gamma^2}$), and Gaussian functions ($\phi(dist) = exp^{-\gamma dist^2}$) where $dist$ denotes the distance between the vector to be measured and the sample vector. The most used basis function for surrogate modeling is the Gaussian function.

$$
RBFN(x) = \sum_{i=1}^{n} \lambda_i \phi\left(dist\left(x - x_i\right)\right) \tag{2}
$$

Support vector regressor (SVR)

Support Vector Regression (SVR) is the regression variant of Support Vector Machines (SVM) [11–13]. SVR mathematically represents a weighted sum of basis functions or kernels (ψ) and a constant term (μ) [10]. Eq. (3) provides the mathematical description of the SVR where ω_i represents the weights for each kernel.

$$
SVR(x) = \mu + \sum_{i=1}^{n} \omega_i \psi\left(dist\left(X - X_i\right)\right) \tag{3}
$$

Eq. (4) expresses Eq. (3) in matrix form where the kernel ψ is replaced by the matrix X.

$$
SVR(x) = \mu + \omega^T X \tag{4}
$$

This surrogate model is comparable to the RBF model. However, the approach employed to compute the unknown parameters in the SVR diverges considerably from the RBF model. The unknown parameters of this model (μ and ω) are obtained by formulating a mathematical optimization problem utilizing Eq. (5).

$$
\begin{aligned}
Minimize : \ & \frac{1}{2}|\omega|^2 + C\frac{1}{n}\sum_{i=1}^{n}\left(\eta^{+(i)} + \eta^{-(i)}\right) \\
& Subject\ to : \\
& \omega \cdot x_i + \mu - y_i \leq \epsilon + \eta^{-(i)}0 \\
& y_i - \omega \cdot x_i - \mu \leq \epsilon + \eta^{+(i)} \\
& \eta^{+(i)}, \eta^{-(i)} \geq 0
\end{aligned} \tag{5}
$$

The first and second constraints of Eq. (5) help the sample vector to remain within a $\pm\epsilon$ deviation of the value at the sample vectors without impacting the surrogate model. This allowable deviation range is called the insensitive tube ϵ. On the contrary, the slack variables $\eta^{+(i)}$ and $\eta^{-(i)}$ serve to ensure the feasibility of the problem by allowing outliers to remain outside the insensitive tube ϵ. The optimal balance between model complexity and fit is obtained by penalizing outliers using a pre-established constant $C \geq 0$. Eq. (5) minimizes the integrated effect of complexity and the outlier penalty. However, finding μ and ω requires solving the quadratic programming problem, which affects the model construction time.

Metrics for evaluating surrogate models' performance

Mean square error (MSE). Mean Square Error (MSE) calculates the average of the square differences between predicted and observed values in regression [14]. Eq. (6) expresses this metric, where V_i and $\hat{V}_i$ are the i-th observed and predicted value.

$$
MSE = \frac{1}{N}\sum_{i=1}^{N}\left(V_i - \hat{V}_i\right)^2 \tag{6}
$$

If the MSE equals zero, the estimator can accurately predict the parameter's response. The quality of the surrogate model improves as the MSE value decreases.

Modified index of agreement (d_j). The modified index of agreement (d_j), also known as the Willmott index, is a standardized measure of prediction ability in regression models [15]. In contrast to the coefficient of determination (R^2), d_j does not allow negative values it ranges from 0 to 1, where zero indicates complete disagreement between the observed and predicted values, and one indicates complete agreement. d_j is defined in Eq. (7), where V_i and $\hat{V}_i$ represent the observed and predicted values, respectively, n is the number of elements in V, and $\bar{V}$ represents the average of the observed values. The j index is usually defined with a value of one.

$$d_j = 1 - \frac{\sum_{i=1}^{n} \left| V_i - \hat{V}_i \right|^j}{\sum_{i=1}^{n} \left| \hat{V}_i - \bar{V} \right| + \left| V_i - \bar{V} \right|^j} \tag{7}$$

Alignment-based similarity measures for time series

Dynamic time warping (DTW). Dynamic Time Warping (DTW) is a method of matching two sequences or time series by expanding or contracting the time component [10,16]. DTW is widely used in speech processing, machine learning, and speech recognition because it resists outliers and can efficiently match two time series with different lengths or phases but similar shapes.

The DWT's basic idea is to find the optimal match or the deformation path utilizing a dynamic programming technique. Let $Ts_1 = ts_1^1, ts_2^2, \ldots, ts_m^1$ and $Ts_2 = \{ts_1^2, ts_2^2, \ldots, ts_n^2\}$ time series, the deformation path is computed by a distance matrix $Dist$ among them, where each cell $dist(ts_i^1, ts_j^2)$ represents the distance between points ts_i^1 and ts_j^2. In this way, the optimal deformation path between two varying-sized and irregular-sampled time series is represented by Eq. (8), where w_z represents the element z^{th} of the deformation path W of size $K \leq n + m - 1$.

$$DTW\left(Ts_1, Ts_2\right) = \min_{W} \sum_{z}^{Z} dist\left(w_z\right) \tag{8}$$

Dynamic programming is employed to determine the most efficient deformation path. Eq. (9) states the procedure for computing this path and, as a result, the DTW distance. $Dist(i, j)$ represents the total distance, including the current distance ($dist(ts_i^1, ts_i^2)$) and the minimum cumulative distances of the three adjacent elements ($DTW(i, j-1)$, $DTW(i-1, j-1)$, and $DTW(i-1, j)$).

$$\begin{aligned} Dist(i,j) &= dist\left(ts_i^1, ts_i^2\right) + adj(i,j) \\ adj(i,j) &= min\begin{cases} DTW(i, j-1) \\ DTW(i-1, j-1) \\ DTW(i-1, j) \end{cases} \end{aligned} \tag{9}$$

However, computing the distance using DTW involves significant computational complexity. Several strategies are suggested to minimize the complexity of DTW. One successful approach is to incorporate constraints that consider only a portion of possible alignments between two varying-sized and irregularly sampled time series. Sakoe and Chiba [17] proposed an efficient global constraint for this objective. Eq. (10) incorporates the Sakoe-Chiba band δ_z into the optimal deformation path calculation described in Eq. (8), where WP is the warping path. This band allows for a more efficient and accurate path computation method for DTW.

$$\begin{aligned} DTW(X,Y) &= \min_{W} \sum_{z=1}^{Z} \delta_z dist\left(w_z\right) \\ \delta_z &= \begin{cases} 1, \delta_z \in WP \\ \infty, \delta_z \notin WP \end{cases} \end{aligned} \tag{10}$$

Global alignment kernel (GAK). Although DTW is a competitive distance measure for time series of different sizes and irregularly sampled, it cannot be used to construct a kernel-based regression method because it does not meet Mercer's condition to be a kernel positive definite [18]. This condition is essential to build kernels since it guarantees that the learning methods converge towards a globally optimal solution [19]. For this reason, several kernel methods have been proposed for machine learning tasks.

The Global Alignment Kernel (GAK) calculates the soft minimum costs of every conceivable alignment [20]. Eq. (11) expresses the computation of the GAK where w and W are similar to Eq. (8). However, Cuturi et al. [18] demonstrated that GAK is a positive semidefinite kernel with a $O(n^2)$ complexity comparable to DTW.

$$\begin{aligned} k_{GA} &= \sum_{w \in W} e^{Dist(w)} \\ &\sum_{w \in W} \prod_{i=1}^{|w|} \psi\left(w_i^1, w_i^2\right) \\ k &= e^{||x-y||^2} \end{aligned} \tag{11}$$

Some strategies employed in DTW can also accelerate GAK computations. Cuturi et al. [21] added the triangular parameter to the GAK method as an additional constraint. This modified version, the Triangular Global Alignment Kernel (TGAK), is presented in

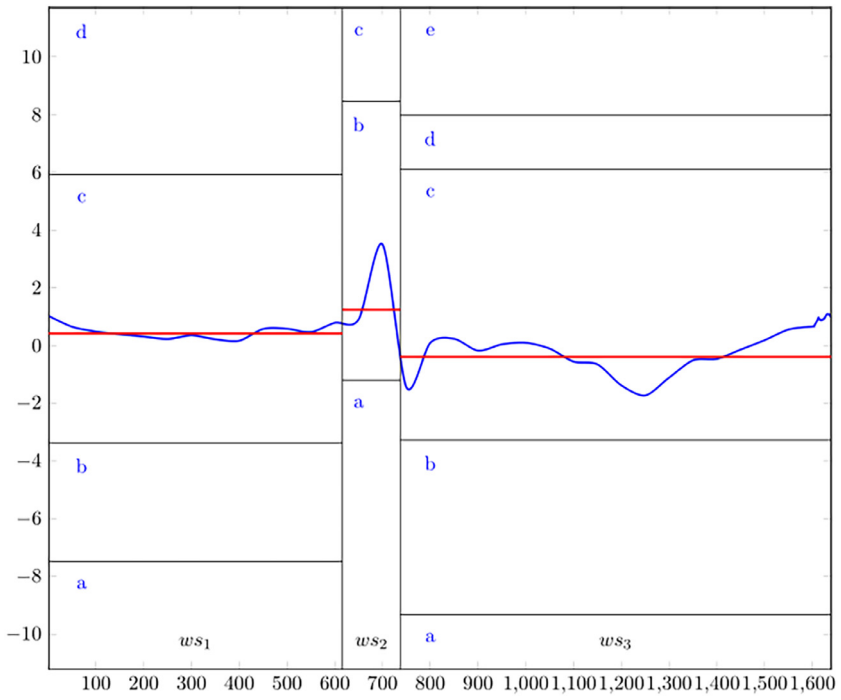

Fig. 1. Example of an eMODiTS discretization scheme where each word segment ws_i contains its own breakpoints scheme. In this example, the final string is cbc.

Eq. (12), where ψ_σ is the Gaussian kernel and T_k is the order of the Toeplitz kernel τ. The region near the diagonal indicates where the optimal deformation path can be found. This process effectively reduces the number of computations required.

$$k_{TGA} = \frac{\tau(i,j)\,\psi_\sigma(x_i,y_i)}{2-\tau(i,j)\,\psi_\sigma(x_i,y_i)}$$
$$\tau(i,j) = max\left[\left(1-\frac{|i-j|}{T_k}\right),0\right] \tag{12}$$

eMODiTS

eMODiTS is a method of symbolically discretizing data by finding the appropriate discretization schemes using a multi-objective evolutionary algorithm (MOEA), NSGA-II [1,2,22]. MOEAs are designed to identify solutions that optimize or minimize two or more functions in compromise with each other. The MOEA interest problem is the multi-objective optimization problem (MOOP). It is expressed as Eq. (13), where $\vec{F}$ is the objective function set to optimize, $\vec{x}$ is the vector of decision variables, m is the number of objective functions, $g(\vec{x})$ and $h(\vec{x})$ are unequal and equal constraint functions.

$$\begin{aligned} Min/Max\ \vec{F} &= \{f_1(\vec{x}),\ldots,f_m(\vec{x})\} \\ &Subject\ to: \\ &g(\vec{x}) \le 0 \\ &h(\vec{x}) \le 0 \end{aligned} \tag{13}$$

MOEAs obtain a set of potential solutions based on Pareto dominance. A vector $\vec{x}_1$ dominates $\preceq$ a vector $\vec{x}_2$ if and only if, $\vec{x}_1 \le \vec{x}_2, \forall_i \in \{1,\cdots,a\} \wedge \vec{x}_1 < \vec{x}_2, \exists_i \in \{1,\cdots,a\}$. The set of non-dominated solutions is called the Pareto front F.

eMODiTS introduced a discretization scheme in which each cut in the time axis (called the word segment) contains its own set of cuts in the value space of the time series (called the set of alphabet cuts or set of breakpoints). Fig. 1 shows an example of an *eMODiTS* scheme.

For the codification of a discretization scheme I in *eMODiTS*, a vector with cuts of word segment followed by its respective alphabet is defined. Eq. (14) expresses an individual's representation in *eMODiTS*, and Fig. 2 illustrates graphically a discretization scheme I, ws_i represents the cut-offs for a word segment and $\alpha_{ws_i,j}$ its respective cut-offs of the alphabet.

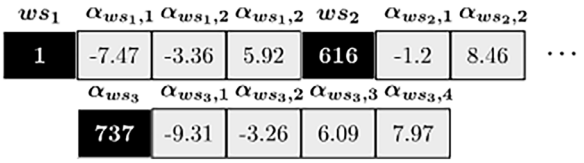

Fig. 2. Codification of an individual in eMODiTS, where ws_i denotes the word segment cut and $\alpha_{ws_i,j}$ expresses its alphabet cuts.

Algorithm 1 Adapted one-point crossover operator procedure.

global variables
  ws_{min}, minimum number of word segments
  ws_{max}, maximum number of word segments
end global variables
procedure OnePointCrossover(Par_1: First parent, Par_2: Second parent)
variables
  cut_1, random cut in time length of Par_1
  cut_2, random cut in time length of Par_2
  Off_1, first offspring
  Off_2, second offspring
  Par_1^{1st}, first part of the first parent
  Par_2^{1st}, first part of the second parent
  Par_1^{2nd}, second part of the first parent
  Par_2^{2nd}, second part of the second parent
end variables
Generate cut_1 and cut_2
$Off_1 \leftarrow \{Par_1^{1st} Par_2^{2nd}\}$
$Par_1^{1st} \leftarrow \{ws_i^{Par_1} A_{ws_i}^{Par_1}\}_1^{cut_1}$
$Par_2^{2nd} \leftarrow \{ws_j^{Par_2} A_{ws_j}^{Par_2}\}_{cut_1}^{|ws^{Par_2}|}$
$Off_2 = \{Par_2^{first} Par_1^{2nd}\}$
$Par_2^{first} = \{ws_j^{Par_2} A_{ws_j}^{Par_2}\}_1^{cut_2}$
$Par_1^{2nd} = \{ws_i^{Par_1} A_{ws_i}^{Par_1}\}_{cut_1}^{|ws^{Par_1}|}$
If $|ws^{Off_1}| \leq ws_{min} \wedge |ws^{Off_2}| \geq ws_{max}$ **then**
  return OnePointCrossover(Par_1,Par_2)
else
  return Off_1 and Off_2
end if
end procedure

$$I = \left\{ ws_1, \mathcal{A}_{ws_1}, ws_2, \mathcal{A}_{ws_2}, \ldots, ws_d, \mathcal{A}_{ws_d} \right\}$$
$$\mathcal{A}_{ws_i} = \left\{ \alpha_{ws_i,1}, \alpha_{ws_i,2}, \ldots, \alpha_{ws_i,j} \right\} \tag{14}$$

In particular, each individual in *eMODiTS* is a high-dimensional vector of varying size. Therefore, we use the popular one-point crossover operator for generating offspring, which has been modified to deal with the problem of varying-sized individuals. Adapting the crossover includes determining a random segment at the word level and combining its sections to produce new offspring. Partitioning is done only at the level of the word segment to avoid inconsistent discretization schemes in which the part of an individual starts with alphabet cuts or ends with word segment cuts. Algorithm 1 describes the adapted crossover operator.

eMODiTS employs three objective functions to find a competitive discretization scheme to classify time series. These functions are Entropy, Complexity, and Information Loss. Eq. (15) expresses each fitness function, respectively, where $\overline{S_i}$ is a set of discrete time series $\overline{S_i}$, $M(P(\overline{S_i}, c_j))$ represents the matrix showing the probabilities of a class being part of a discrete time series, D expresses the original temporal dataset, R is the reconstructed temporal data set. Entropy (E) estimates the class separability in the discretized temporal dataset, complexity (Cx) estimates how complex the discrete space is found in terms of the number of unique discrete time series, and information loss (IL) estimates the amount of data loss due to dimensionality reduction. It is essential to mention that the original (Ts_i) and reconstructed time series (Tr_i) have been scaled to [0,1] to facilitate a fair comparison between the two temporal data.

$$E = \sum_{i=1}^{|\bar{S}|} - \sum_{j=1}^{|C|} M\left(P\left(\overline{S_i}, c_j\right)\right) log_2 M\left(P\left(\overline{S_i}, c_j\right)\right)$$
$$Cx = \begin{cases} \frac{|T|-|\bar{S}|-|C|}{|T|+(|C|-1)}, & |\bar{S} - |C| < 0| \\ \frac{|\bar{S}|-|C|}{|T|+(|C|-1)}, & otherwise \end{cases}$$

Algorithm 2 eMODiTS overall procedure.

Procedure *eMODiTS*(PS: Population size, G: number of generations, pm: mutation rate, pc: crossover rate)
variables
  P, current population
  F, Pareto front
  O, set of offsprings

end variables
Randomly create P of size PS
Evaluate P using Equation
For $g = 1, 2, \ldots, G$ **do** *//NSGA-II stage*
  Rank P based on NSGA-II non-dominated sorting. Store it in F
  Compute the crowding distance of F
  Select parents using the selection mechanism described in NSGA-II
  Create offspring set O using Algorithm 1
  Mutate offspring based on a uniform distribution, randomly changing a value in each individual of P
  Evaluate the offspring using Eq. (15)
  $P' \leftarrow \{PO\}$
  Rank P' based on the fast nondominated sorting of NSGA-II method. Store it in F
  Compute the crowding distance of F
  Replace P with the first fronts of F
End for
Return F
end procedure

$$IL = \frac{1}{|R|}\sum_{i=1}^{|R|}\left[\frac{1}{n-1}\sum_{j=1}^{n}\left(Tr_i - Ts_j\right)^2\right] \tag{15}$$

NSGA-II was utilized as a search algorithm to identify appropriate data-driven discretization methods using the above-mentioned elements and keeping the canonical procedures of NSGA-II. The general procedure of *eMODiTS* is described in Algorithm 2.

asMODiTS

The literature reports the application of surrogate models for time series classification [23–25]. However, these proposals are based on training sets with fixed-size instances.

Some time series classification approaches propose methods for handling training set instances of variable size. One of the main approaches used is DTW [26–28] due to its ability to search for similar subsequences between two-time series, obtaining the similarity between both series. Another approach uses Shapelets, which are discriminative subsequences of time series data. Shapelet-based methods extract shapelets from time series and use them to classify varying-length time series [29–31]. They are effective for capturing local patterns.

On the other hand, some authors employ an algorithm to convert time series into fixed-length vectors by quantizing and histogram calculation of subsequences [16,32,33]. This method is called the Bag-of-Words (BoW) Model and is widely used in natural language processing.

The *archive-based and surrogate-assisted multi-objective approach for symbolic discretization of time series (asMODiTS)* is a method in which surrogate models were implemented to decrease the computational cost of discretization and evaluation of each objective function of *eMODiTS*. Therefore, KNN, RBFN, and SVR are proposed as surrogate models. Since each individual in the *eMODiTS* population presents different sizes, KNN and RBFN are implemented using DTW as a distance measure, while SVR uses the GAK. Moreover, an external archive Λ of size $PS/4$ is also included to maintain diversity in the training set and increase the fidelity of the surrogate model.

On the other hand, a hybrid approach for the evolution control stage is proposed to update the surrogate models. This mechanism consists of applying the fixed and dynamic control evolution together. The fixed approach performs the generation-based strategy every GU generation per surrogate model, replacing the individual quality of the current population with the original value and storing the farthest individuals from the training set within the archive. The training set of each surrogate method is updated using the TI individuals in the archive with the highest prediction error based on an error threshold ε. Furthermore, IU individuals in the current population are evaluated using the original objective functions in each generation (individual-based strategy).

After the fixed control evolution, GU is updated according to the percentage of individuals inserted into the training set multiplied by a penalty factor π. It is the dynamic approach for the control evolution step.

The Algorithm 3 describes the incorporation of the surrogate model and how it was updated.

Algorithm 3 asMODiTS overall procedure where surrogate models are incorporated for evaluating new solutions.

Procedure *asMODiTS*(PS: Population size, G: number of generations, pm: mutation rate, pc: crossover rate,$\tilde{m}$: surrogate model, $\tilde{m}_{GU}$: number of generations where the surrogate model is updated, TI: number of individuals from the archive used to update the training set, IU: number of individuals from the current population to evaluate in the original functions, Π: factor to calculate the size of the training set, ξ: threshold to update the training set).
  variables
    P, current population
    $\tilde{m}_{i,train}$, training set of the surrogate model i
    F, Pareto front
    O, set of offsprings
    Λ, the external and auxiliary archive used to maintain diversity in $\tilde{m}_{i,train}$
    ϵ, estimation error between the value of the surrogate model and the original functions
    L, archive maximum size
    m, Number of objective functions
    $inserted$, number of individuals inserted in the training set based on its estimation error
    ρ, Percentage of individuals inserted in the training with respect to TI
  end variables
  Randomly create a training set of size $PS * \Pi$ and store it in each $\tilde{m}_{i,train}$
  Create P randomly extracting PS individuals from the training set
  Evaluate P and $\tilde{m}_{i,train}, i = \{1, \cdots, m\}$ in the original functions (Eq. (15))
  Train $\tilde{m}_i, i = \{1, \cdots, m\}$
  Set $L = PS/4$
  For $g = 1, 2, \ldots, G$ **do** *//NSGA-II stage*
    Rank P based on NSGA-II non-dominated sorting. Store it in F
    Compute the crowding distance of F
    Select parents using the selection mechanism described in NSGA-II
    Create offspring set O using Algorithm 1
    Mutate offspring based on a uniform distribution, randomly changing a value in each individual of P
    Evaluate the offspring in $\tilde{m}_i, i = \{1, \cdots, m\}$
    Store the offspring farthest from the training set in Λ
    Prune Λ when it exceeds L
    Evaluate IU individuals from P in original functions (Eq. (15)) //Individual-based strategy
    $P' \leftarrow \{PO\}$
    Rank P' based on the fast nondominated sorting of NSGA-II method. Store it in F
    Compute the crowding distance of F
    Replace P with the first fronts of F
    For $i = 1 \cdots$ m **do**
      If $(g\%\ \tilde{m}_{i_{GU}}) = 0$ // Generation-based strategy
    If $|\Lambda| < TI$
      Insert $TI - |\Lambda|$ new individuals randomly generated in $\tilde{m}_{i,train}$
    else
      Select TI individuals from Λ
      Evaluate the selected individuals in the original functions (Eq. (15))
      For $k = 1 \ \ldots TI$ **do**
        Set ind as the selected individual k
        Calculate the estimation error ϵ of ind using the measure d_j (Eq. (7))
        $inserted = 0$
        If $\epsilon > \xi$
          Insert ind in $\tilde{m}_{i,train}$
          Train $\tilde{m}_i$
          $inserted = inserted + 1$
        End if
      End for
      $\rho = inserted/TI$
      $\tilde{m}_{i_{GU}} = round(\tilde{m}_{i_\pi} * \rho, 0)$ //Dynamic evolution control
    End if
      Evaluate P in the original functions (Eq. (15)) //Generation-based strategy
    End if
    End if
  End for
  Return F
end procedure

Method validation

Input datasets

The data sets used to validate our method were obtained from the UCR repository [34]. This repository is an open-access resource to evaluate the effectiveness of the proposed methods for time series classification. A total of 17 data sets were selected to test the

Table 1
Description of the datasets used for the analysis in this study. The data were extracted from the UCR Repository [22]. Abbreviation column (Abbrev.) was incorporated by authors for further references. TrS, TeS, L, NoC, and T denotes Training S.

Dataset	Abbrev.	TrS	TeS	L	NoC	T
CBF	CBF	30	900	128	3	SIMULATED
DistalPhalanxOutlineAgeGroup	DPOAG	400	139	80	3	IMAGE
ECG200	ECG200	100	100	96	2	ECG
ECGFiveDays	ECG5D	23	861	136	2	ECG
FaceAll	FALL	560	1690	131	14	IMAGE
FacesUCR	FUCR	200	2050	131	14	IMAGE
ItalyPowerDemand	ITAPD	67	1029	24	2	EPG
MedicalImages	MEDIMG	381	760	99	10	HAR
MiddlePhalanxOutlineAgeGroup	MPOAG	400	154	80	3	SENSOR
MiddlePhalanxTW	MPTW	399	154	80	6	OTHER
MoteStrain	MOTEST	20	1252	84	2	SIMULATED
ProximalPhalanxTW	PPTW	400	205	80	6	SENSOR
SonyAIBORobotSurface1	SONY1	20	601	70	2	IMAGE
SonyAIBORobotSurface2	SONY2	27	953	65	2	MISC
SwedishLeaf	SWEDLF	500	625	128	15	SOUND
SyntheticControl	SYNCTR	300	300	60	6	SENSOR
TwoPatterns	TWOPAT	1000	4000	128	4	SENSOR

Table 2
Parameter setting obtained by the Bayesian optimization method proposed in [23].

Surrogate model	Objective function	Parameter	Value
KNN	Entropy	Sakoe-Chiba Window	0.4315
		k	1
	Complexity	Sakoe-Chiba Window	0.5889
		k	5
	Information Loss	Sakoe-Chiba Window	0.8808
		k	1
SVR	Entropy	Sakoe-Chiba Window	0.8199
		C	10
	Complexity	Sakoe-Chiba Window	0.3703
		C	10
	Information Loss	Sakoe-Chiba Window	0.9392
		C	9
RBFN	Entropy	Sakoe-Chiba Window	0.8292
		Epoch	50
		k	8
		Learning rate	0.099
	Complexity	Sakoe-Chiba Window	0.4512
		Epoch	100
		k	8
		Learning rate	0.0915
	Information Loss	Sakoe-Chiba Window	0.4652
		Epoch	150
		k	7
		Learning rate	0.1676

robustness of our method due to the properties they present (different numbers of classes, data dimensionality, and data type) [2]. See Table 1 for further details.

Parameter setting

The parameters used in *asMODiTS* are the same as the original method (*eMODiTS*) for a fair comparison (PS, G, pm, pc). The values for the parameters GU, TI, and IU were obtained based on previous experiments. The experiments were conducted to test the efficacy of our method under various configurations of values. The values used for GU were every 30, 60, and 100 generations; for IU, we used 5 %, 10 %, and 30 %; for TI, we used 1, 10, and 50 individuals. The optimal values were GU=60, IU=10 %, and TI=10. However, the parameter values of each surrogate model were obtained using the Bayesian Optimizer implemented in Python [35]. The selected optimizer has been demonstrated to identify optimal hyperparameter configurations for machine learning algorithms in various contexts [36–38]. This optimization used 200 evaluations, d_j as the evaluation measure, and the training set proposed in [34] for each data set. Furthermore, this set was subdivided using 50 % of the instances for the training set and 50 % for the testing set during optimization. The values obtained by the optimizer are described in Table 2.

Table 3
Prediction power of each surrogate model in each dataset calculated through d_j metric. $asKNN$, $asRBFN$, and $asSVR$ denote the $asMODiTS_{KNN}$, $asMODiTS_{RBFN}$, and $asMODiTS_{SVR}$ methods, respectively. Bold numbers represent the highest values per data set for each surrogate model. Moreover, $\bar{X}$, $\tilde{X}$, and σ indicate the average, median, and standard deviation of data.

Dataset	Entropy			Complexity			Information Loss		
	$asKNN$	$asRBFN$	$asSVR$	$asKNN$	$asRBFN$	$asSVR$	$asKNN$	$asRBFN$	$asSVR$
CBF	0.1682	**0.3796**	0.1733	**0.5968**	0.468	0.4753	0.7794	**0.8235**	0.4482
DISTAG	0.0243	0.0026	**0.0765**	**0.6851**	0.3569	0.1816	**0.5103**	0.4339	0.3995
ECG200	**0.0277**	0.0275	0.0215	**0.5742**	0.2367	0.2947	**0.8476**	0.4763	0.2559
ECG5D	0.1002	0.1081	**0.2062**	0.4519	**0.5136**	0.3827	**0.392**	0.3394	0.3494
FALL	0.0067	0.0106	**0.0588**	**0.5966**	0.3845	0.4454	**0.5946**	0.2312	0.488
FUCR	0.0141	0.0364	**0.2214**	**0.517**	0.4887	0.3791	**0.5309**	0.2465	0.3683
ITAPD	0.1398	0.2217	**0.941**	0.494	0.4464	**0.8601**	0.5277	0.3277	**0.9692**
MEDIMG	0.0126	0.2355	**0.2629**	**0.7108**	0.3706	0.5806	**0.706**	0.1655	0.2636
MIDAG	0.0072	0.0047	**0.0112**	**0.5118**	0.2557	0.3823	**0.575**	0.477	0.3414
MIDTW	0.0075	**0.1813**	0.047	**0.536**	0.5029	0.4155	**0.4115**	0.3944	0.4106
MOTEST	0.1988	0.1525	**0.2691**	**0.6473**	0.4171	0.2704	**0.7369**	0.2939	0.4382
PROXTW	0.008	0.078	**0.134**	0.5397	0.2212	**0.5427**	**0.4785**	0.4561	0.2254
SONY1	0.2706	0.208	**1**	0.5463	0.4893	**1**	0.6571	0.6177	**1**
SONY2	0.2208	**0.6831**	0.3451	**0.796**	0.5919	0.4639	0.6779	**0.8852**	0.4232
SWEDLF	0.0055	0.0021	**0.0754**	**0.5502**	0.3895	0.3453	0.547	**0.5512**	0.3369
SYNCTR	0.0218	0.0075	**0.0825**	**0.675**	0.3633	0.3857	**0.7886**	0.4771	0.3874
TWOPAT	0.003	0.0134	**0.1141**	**0.6894**	0.3149	0.3825	**0.7688**	0.5771	0.399
$\bar{X}$	0.0728	0.1384	**0.2376**	**0.5952**	0.4007	0.4581	**0.6194**	0.4573	0.4414
$\tilde{X}$	0.0218	0.078	**0.134**	**0.5742**	0.3895	0.3857	**0.5946**	0.4561	0.399
σ	**0.0908**	0.1784	0.2919	**0.0919**	0.1044	0.203	**0.1382**	0.1952	0.2162

Preliminary results and discussions

The experiments performed in this study were designed to validate our method in terms of the surrogate model's fidelity and the classification task's precision.

Three surrogate models were implemented to analyze its behavior in estimating the objective functions: KNN with DTW as distance measure ($asMODiTS_{KNN}$), SVR with GAK as distance measure ($asMODiTS_{SVR}$), and RBFN with DTW as distance measure ($asMODiTS_{RBFN}$). Table 3 shows the results of the prediction power calculated with the metric d_j.

As can be seen, the information loss function obtained the highest prediction values, reaching values above 0.6. Meanwhile, the entropy function obtained the worst prediction power with values greater than 0.2. Regarding the behavior of each method, $asMODiTS_{KNN}$ obtained the highest prediction values in both the complexity function and the information loss function, being the method with the best performance. On the other hand, $asMODiTS_{SVR}$ was the method that had the best prediction performance in most databases concerning the entropy function. These results are confirmed by the mean ($\bar{X}$), median ($\tilde{X}$), and standard deviation (σ), where we can see for each function which method outperforms the others. In the same way, Fig. 3 graphically shows the behavior of the predictive power of each method in each function, where the information loss function achieved the more accurate prediction compared to the others. The $asMODiTS_{KNN}$ method was the one that best approximated the values of each function.

Predictive values reported by our procedure could result from training set coding due to the high dimensionality of the training set and the different sizes of each in the same set. These characteristics are usually challenging to handle by most regression algorithms, making it difficult to learn the main patterns present in the data [39,40]. In addition, high-dimensional data often present a high risk of overfitting in the training process, leading to poor generalization in the presence of new data [41].

Regarding the classification task, the surrogate models were compared with the original method. For this purpose, the F1 score was used due to its insensitivity to class imbalance, a problem that affects other metrics, such as precision. Table 4 shows the results obtained by each approach using this metric, where the values in bold represent the highest values of the F1 score, representing a competitive classification of the data set classes. This table shows that *eMODiTS* with the original models obtain competitive results in more temporal data sets than the surrogate models.

A nonparametric statistical test to validate our analysis was employed because the results did not follow a normal distribution. Friedman's multi-comparison test and Nemenyi's post hoc test were used with a confidence level of 95 %. The results are shown in Fig. 4 and confirm the above. $eMODiTS$ ranked highest, while $asMODiTS_{SVR}$, $asMODiTS_{RBFN}$, and $asMODiTS_{KNN}$ followed. However, there are no significant statistical differences between the different approaches. Thus, it can be concluded that they exhibit similar behavior in classifying time series. Such results are expected when a surrogate model replaces computationally expensive parts of the multi-objective optimization process. In simpler terms, a desirable outcome of surrogate models is to find similar but not superior results. Such models are unlikely to improve the original results, but they will match them.

Finally, Fig. 5 illustrates the computational cost reached by $eMODiTS$, $asMODiTS_{KNN}$, $asMODiTS_{RBFN}$, $asMODiTS_{SVR}$. The computational cost was measured based on the number of evaluations performed by each method in the original objective function.

This figure illustrates that the surrogate methods conducted the fewest evaluations of the original model. It is important to note that evaluating the original objective functions involves discretizing the time series to calculate them. Therefore, the estimation performed

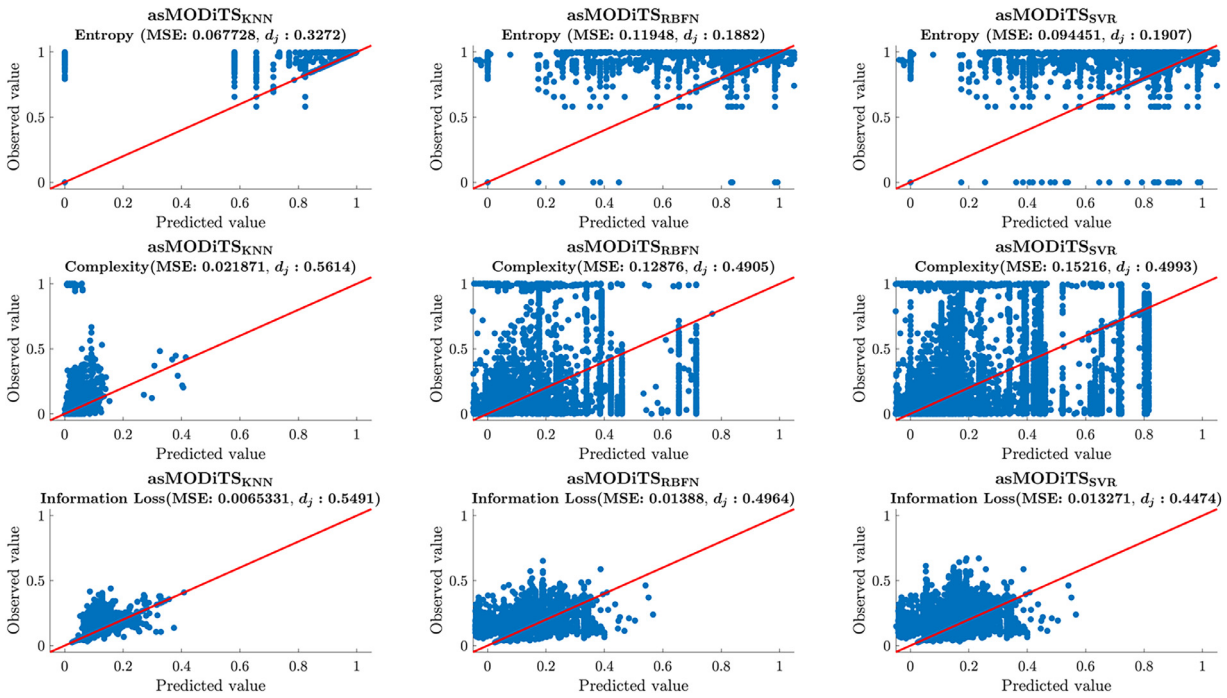

Fig. 3. Predicted and actual plot for $asMODiTS_{KNN}$, $asMODiTS_{RBFN}$, $asMODiTS_{SVR}$.

Table 4
Classification accuracy results reached by the original eMODiTS and its surrogate versions ($asMODiTS_{KNN}$, $asMODiTS_{RBFN}$, $asMODiTS_{SVR}$). Dataset column shows the abbreviation column of the Table 1. F1 score and the standard deviation were employed to express the results. Bold numbers represent the higher F1 scores obtained per dataset. Values after ± represent the standard deviation of each execution.

Abbrev.	$eMODiTS$	$asMODiTS_{KNN}$	$asMODiTS_{RBFN}$	$asMODiTS_{SVR}$
CBF	0.8222±0.013	0.8501±0.0082	0.8731±0.0134	**0.8927±0.0037**
DISTAG	0.7921±0.0085	**0.825±0.0153**	0.8123±0.0199	0.8183±0.0116
ECG200	**0.8326±0.0394**	0.7975±0.0196	0.7651±0.0165	0.792±0.0175
ECG5D	0.773±0.0236	**0.7936±0.0254**	0.6896±0.013	0.6671±0.0176
FALL	**0.6899±0.0091**	0.5076±0.0095	0.4892±0.0102	0.6515±0.0064
FUCR	0.5435±0.0076	0.3313±0.0107	0.5625±0.0101	**0.6601±0.0117**
ITAPD	0.938±0.0036	0.9589±0.0027	0.9617±0.0032	**0.9653±0.0036**
MEDIMG	**0.6494±0.0071**	0.4596±0.0049	0.627±0.0101	0.5916±0.0072
MIDAG	0.737±0.026	0.7463±0.0274	0.7395±0.0189	**0.7606±0.0231**
MIDTW	0.5499±0.0089	**0.5645±0.0112**	0.5551±0.0123	0.5368±0.011
MOTEST	0.7169±0.0054	0.7239±0.0157	**0.7804±0.0077**	0.745±0.0082
PROXTW	0.7593±0.017	0.7319±0.0064	0.7195±0.0143	**0.7679±0.0163**
SONY1	0.8076±0.0141	0.5522±0.0132	**0.8257±0.0113**	0.4044±0.0039
SONY2	**0.811±0.0119**	0.7546±0.0137	0.7945±0.0135	0.7718±0.0194
SWEDLF	**0.6312±0.024**	0.488±0.0047	0.569±0.0142	0.5757±0.0076
SYNCTR	**0.923±0.0143**	0.8446±0.027	0.8296±0.0209	0.8853±0.0212
TWOPAT	**0.8177±0.002**	0.6111±0.0065	0.5666±0.0035	0.5557±0.0059

by each surrogate method implies obtaining the values of the objective functions without performing the discretization process, minimizing the main computational cost of $eMODiTS$. In summary, the surrogate methods significantly reduced the computational cost of the original method. However, using a costly distance measure (DTW) resulted in longer execution times than $eMODiTS$ due to the coding of the individuals in the training set.

New research directions

The results obtained from our method allow us to identify several future research directions.

- The predictive power of surrogate models can be analyzed by changing how individuals are represented in the training set. This analysis could involve proposing more straightforward representations with high fidelity to the original model.

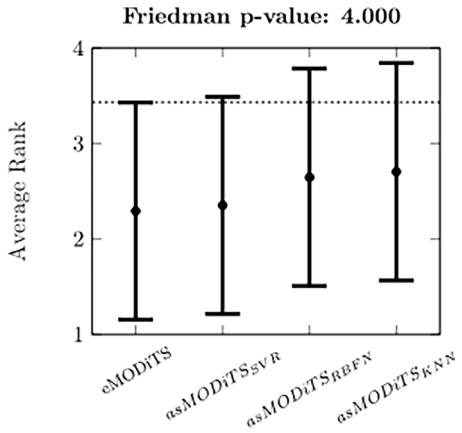

Fig. 4. Statistical test results from the Friedman test with Nemenyi post hoc at 95 %-confidence.

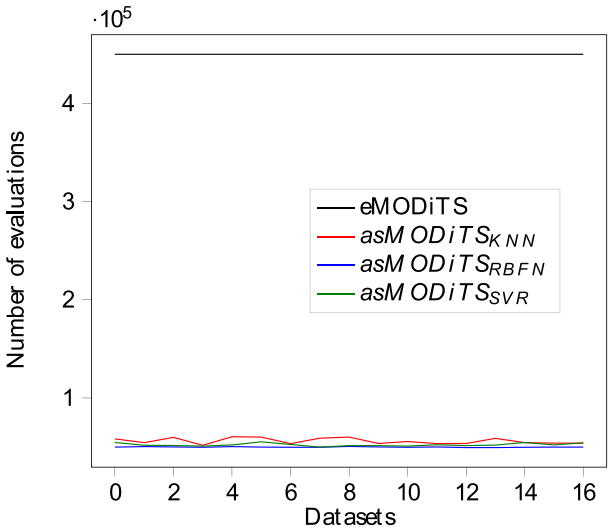

Fig. 5. Computational cost measured as the number of evaluations of each method.

- Other surrogate models could be explored to increase their fidelity to the original. This analysis can use more complex models that capture the original functions' behavior, such as neural networks, random forests, and hybrid models.
- Examine the behavior of several subrogated models for each objective function individually and thus select one for each objective function. This future direction aims to implement subrogated models that are ad hoc to the behavior of each objective function. This proposal would enhance the predictive capacity of the proposed method.
- Finally, the proposal's efficacy will be analyzed by utilizing high-performance computing to minimize the inherent computational cost of the DTW algorithm.

Conclusions

This research presents a surrogate method to minimize the computational cost of the discretization approach called $eMODiTS$. Therefore, three approximation models were implemented to approximate the values of each objective function of $eMODiTS$. These models were KNN ($asMODiTS_{KNN}$), RBFN ($asMODiTS_{RBFN}$), and SVR ($asMODiTS_{SVR}$).

The classification rates achieved by the surrogate methods were similar to those obtained by the original method but with a lower computational cost. However, the surrogate methods' predictive power is less than expected, a significant aspect that needs improvement and represents their main disadvantage. Therefore, our proposal is a promising method for discretizing temporal datasets while minimizing computational costs.

It is crucial to emphasize that the novelty of our work lies in the implementation of surrogate models in time series discretization tasks. To our knowledge, this area has not been extensively explored. This situation generates a new line of research in time series preprocessing. Moreover, most investigations employ surrogate models, which typically utilize representations of the training set with a fixed number of variables. Consequently, this method not only provides a new direction in the field of time mining but also

in any problem where it is desired to minimize the computational cost with surrogate models where instances of the training set of fixed size are not employed.

Moreover, using surrogate models to reduce the computational cost of complex problems may be influenced by intentional and unintentional biases. Therefore, to address this ethical issue, automated search tools were employed in this study to identify optimal parameter values without any authors influencing the results presented, thereby enhancing the transparency of the proposed approach.

As a future work, we propose to explore new representations for the individuals in the training set of the surrogate models to avoid the dimensionality course and the increase of the execution time due to DTW. Moreover, comparing more machine learning methods for estimating the objective function values, such as implementing a Random Forest algorithm and Proximity Forest, among others, will be interesting. Finally, an implementation of DTW is proposed for high-performance computing such as CUDA or OpenGL.

Limitations

Not applicable.

Ethics statements

Not applicable.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Aldo Márquez-Grajales: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft. **Efrén Mezura-Montes:** Conceptualization, Methodology, Validation, Writing – review & editing. **Héctor-Gabriel Acosta-Mesa:** Conceptualization, Writing – review & editing. **Fernando Salas-Martínez:** Formal analysis, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

The first author is grateful to the National Council for Humanities, Science, and Technology (CONAHCYT), for the grant awarded through the CVU with number 419862 for the completion of this project. The authors would like to thank Prof. Eamonn Keogh and all the people who collected, preprocessed, and donated the data for the UCR TSC archive.

References

[1] A. Márquez-Grajales, H.G. Acosta-Mesa, E. Mezura-Montes, M. Graff, A multi-breakpoints approach for symbolic discretization of time series, Knowl. Inf. Syst. 62 (2020) 2795–2834.
[2] A. Márquez-Grajales, E. Mezura-Montes, H.G. Acosta-Mesa, F. Salas-Martínez, H. Calvo, L. Martínez-Villaseñor, H. Ponce, R. Zatarain Cabada, M. Montes Rivera, E. Mezura-Montes, Use of a surrogate model for symbolic discretization of temporal data sets through eMODiTS and a training set with varying-sized instances, in: Advances in Computational Intelligence. MICAI 2023 International Workshop, Springer Nature Switzerland, Cham, 2024, pp. 360–372.
[3] A. Márquez-Grajales, E. Mezura-Montes, H.G. Acosta-Mesa, F. Salas-Martínez, A surrogate-assisted symbolic time-series discretization using multi-breakpoints and a multi-objective evolutionary algorithm, Preprints (2024). 10.20944/preprints202406.1126.v1.
[4] Y. Jin, Surrogate-assisted evolutionary computation: recent advances and future challenges, Swarm Evol. Comput. 1 (2011) 61–70, doi:10.1016/j.swevo.2011.05.001.
[5] M.E. Miranda-Varela, E. Mezura-Montes, Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study, Appl. Soft Comput. 73 (2018) 215–229.
[6] L. Shi, K. Rasheed, A survey of fitness approximation methods applied in evolutionary algorithms, in: Computational Intelligence in Expensive Optimization Problems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 3–28, doi:10.1007/978-3-642-10701-6_1.
[7] R. Goyal, P. Chandra, Y. Singh, Suitability of KNN regression in the development of interaction based software fault prediction models, IERI Procedia 6 (2014) 15–21.
[8] D.S. Broomhead, D. Lowe, Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
[9] Q. Liu, X. Wu, Q. Lin, J. Ji, K.C. Wong, A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion, Swarm Evol. Comput. 60 (2021) 100787, doi:10.1016/j.swevo.2020.100787.
[10] A. Bhosekar, M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: a review, Comput. Chem. Eng. 108 (2018) 250–267, doi:10.1016/j.compchemeng.2017.09.017.
[11] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge university press, 2000.
[12] J. Shawe-Taylor, N. Cristianini, Kernel Methods For Pattern Analysis, Cambridge university press, 2004.
[13] A. Christmann, I. Steinwart, Support Vector Machines, Springer, 2008.
[14] P. Jiang, Q. Zhou, X. Shao, Surrogate Model-Based Engineering Design and Optimization, Springer Nature Singapore, 2020 https://books.google.com.mx/books?id=E1XyzQEACAAJ.

[15] D.R. Legates, G.J. McCabe, Evaluating the use of "goodness-of-fit" Measures in hydrologic and hydroclimatic model validation, Water Resour. Res. 35 (1999) 233–241, doi:10.1029/1998WR900018.
[16] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 262–270.
[17] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. 26 (1978) 43–49, doi:10.1109/TASSP.1978.1163055.
[18] M. Cuturi, J.P. Vert, O. Birkenes, T. Matsui, A kernel for time series based on global alignments, in: Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, 2007 II-413-II-416, doi:10.1109/ICASSP.2007.366260.
[19] Y. Xu, Q. Ye, Generalized Mercer kernels and Reproducing Kernel Banach spaces, American Mathematical Society, 2019.
[20] L. Chen, S. Wan, Intelligent fault diagnosis of high-voltage circuit breakers using triangular global alignment kernel extreme learning machine, ISA Trans. 109 (2021) 368–379, doi:10.1016/j.isatra.2020.10.018.
[21] M. Cuturi, Fast global alignment kernels, in: Proceedings of the 28th International Conference on Machine Learning, Madison, WI, USA, Omnipress, 2011, pp. 929–936.
[22] A. Márquez-Grajales, H.G. Acosta-Mesa, E. Mezura-Montes, R. Hernández-Jiménez, N. Pérez-Castro, A.E. Aguilar-Justo, F. Salas-Martínez, Classification of colposcopic images using a multi-breakpoints discretization approach on temporal patterns, Biomed. Signal Process. Control 69 (2021) 102918, doi:10.1016/j.bspc.2021.102918.
[23] A. Renjini, V. Raj, M.S. Swapna, S. Sreejyothi, S. Sankararaman, Phase portrait for high fidelity feature extraction and classification: a surrogate approach, Chaos Interdiscip. J. Nonlinear Sci. 30 (2020) 113122, doi:10.1063/5.0020121.
[24] X.Y. Sun, D. Gong, S. Li, Classification and regression-based surrogate model-assisted interactive genetic algorithm with individual's fuzzy fitness, in: Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, New York, NY, USA, Association for Computing Machinery, 2009, pp. 907–914, doi:10.1145/1569901.1570025.
[25] H.B. Nguyen, B. Xue, P. Andreae, G. Squillero, K. Sim, Surrogate-model based particle swarm optimisation with local search for feature selection in classification, in: Applications of Evolutionary Computing, Springer International Publishing, Cham, 2017, pp. 487–505.
[26] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1994, pp. 359–370.
[27] M. Cuturi, M. Blondel, Soft-dtw: a differentiable loss function for time-series, in: Proceedings of the International Conference on Machine Learning, PMLR, 2017, pp. 894–903.
[28] Y. Wan, X.L. Chen, Y. Shi, Adaptive cost dynamic time warping distance in time series analysis for classification, J. Comput. Appl. Math. 319 (2017) 514–520.
[29] T. Rakthanmanon, E. Keogh, Fast shapelets: a scalable algorithm for discovering time series shapelets, in: Proceedings of the SIAM International Conference on Data Mining 2013 - SDM, 2013, pp. 668–676, doi:10.1137/1.9781611972832.74. n.d..
[30] G. Li, B. Choi, J. Xu, S.S. Bhowmick, K.P. Chun, G.L.H. Wong, ShapeNet: a shapelet-neural network approach for multivariate time series classification, Proc. AAAI Conf. Artif. Intell. 35 (2021) 8375–8383, doi:10.1609/aaai.v35i9.17018.
[31] W. Yan, G. Li, Z. Wu, S. Wang, P.S. Yu, Extracting diverse-shapelets for early classification on time series, World Wide Web 23 (2020) 3055–3081.
[32] J. Wang, P. Liu, M.F. She, S. Nahavandi, A. Kouzani, Bag-of-words representation for biomedical time series classification, Biomed. Signal Process. Control 8 (2013) 634–644.
[33] Z.W. Gui, Y.R. Yeh, Time series classification with temporal bag-of-words model, in: Proceedings of the Conference on Technologies and Applications of Artificial Intelligence, Springer, 2014, pp. 145–153.
[34] E. Keogh et al., The UCR time series classification archive, (2018).
[35] J. Bergstra, D. Yamins, D. Cox, S. Dasgupta, D. McAllester, Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures, in: Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, PMLR, 2013, pp. 115–123. https://proceedings.mlr.press/v28/bergstra13.html.
[36] J. Snoek, H. Larochelle, R.P. Adams, F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger, Practical bayesian optimization of machine learning algorithms, Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012 https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.
[37] J. Wu, X.Y. Chen, H. Zhang, L.D. Xiong, H. Lei, S.H. Deng, Hyperparameter optimization for machine learning models based on bayesian optimization, J. Electron. Sci. Technol. 17 (2019) 26–40, doi:10.11989/JEST.1674-862X.80904120.
[38] A.H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization, Evol. Syst. 12 (2021) 217–223, doi:10.1007/s12530-020-09345-2.
[39] Y. Wang, Y. Fan, P. Bhatt, C. Davatzikos, High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables, Neuroimage 50 (2010) 1519–1535, doi:10.1016/j.neuroimage.2009.12.092.
[40] C. Cui, D. Wang, High dimensional data regression using Lasso model and neural networks with random weights, Inf. Sci. 372 (2016) 505–517, doi:10.1016/j.ins.2016.08.060.
[41] A. Spooner, E. Chen, A. Sowmya, P. Sachdev, N.A. Kochan, J. Trollor, H. Brodaty, A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction, Sci. Rep. 10 (2020), doi:10.1038/s41598-020-77220-w.