



HAL
open science

RSAST: Sampling Shapelets for Interpretable Time Series Classification

Nicolas Rojas Varela, Michael Franklin Mbouopda, Engelbert Mephu Nguifo

► **To cite this version:**

Nicolas Rojas Varela, Michael Franklin Mbouopda, Engelbert Mephu Nguifo. RSAST: Sampling Shapelets for Interpretable Time Series Classification. 2023. <hal-04311309v3>

HAL Id: hal-04311309

<https://hal.science/hal-04311309v3>

Preprint submitted on 24 Jul 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

RSAST: Sampling Shapelets for Interpretable Time Series Classification

Nicolas Rojas Varela
University Clermont Auvergne,
Clermont Auvergne INP,
CNRS, ENSMSE, INP, LIMOS,
Clermont Ferrand, France
nicolas.rojas_varela@doctorant.uca.fr

Michael Franklin Mbouopda
Yipa Technologies
Toulouse, France
mf.mbouopda@gmail.com

Engelbert Mephu Nguifo
University Clermont Auvergne,
Clermont Auvergne INP,
CNRS, ENSMSE, INP, LIMOS,
Clermont Ferrand, France
engelbert.mephu_nguifo@uca.fr

Abstract—Shapelet-based techniques are widely utilized in time series classification due to their combination of interpretability and accuracy. However, these approaches often face scalability challenges compared to other state-of-the-art (SOTA) techniques, primarily due to the large search space of subsequences in datasets with numerous or large instances. To address this problem, we propose RSAST, a method based on shapelet techniques, specifically SAST and STC. The method employs a stratified approach and specific statistical criteria to reduce the subsequence search space of these techniques. As a result, RSAST significantly decreases computation time while preserving classification performance and interpretability.

Index Terms—Time Series Classification, Shapelet, Scalability, Interpretability, ANOVA, Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF).

I. INTRODUCTION

Time series (TS) data is a type of data collected at different points in time. It is used in different tasks like anomaly detection, forecasting, and classification. Time series classification (TSC) aims to predict the label of a given time series. TSC has applications in various fields, including Human Activity Recognition (HAR) and spectroscopy for food quality assurance, among others.

Several methods for time series classification can be categorized based on the mechanisms they use for classification [12]:

- Distance-based methods: Entire time series are compared using distance measures in combination with nearest neighbor (NN) classifiers.
- Feature-based methods: These methods involve extracting features from the time series and using them in a classifier.
- Interval-based methods: In these approaches, it is selected one or more intervals within a time series rather than using the entire sequence.
- Shapelet-based methods: Algorithms in this category aim to identify patterns or subsequences from a training dataset of TS that define a class. These patterns, known as shapelets, can appear anywhere in the series and are thus phase-independent.
- Dictionary-based methods: These approaches focus on the frequency of repeating subseries rather than their mere

presence or absence. They involve constructing frequency counts of recurring patterns and utilizing the resulting histograms for classification.

- Convolution-based methods: These methods consist of the application of convolution and pooling operations to time series.
- Deep learning-based methods: This category utilizes neural network structures for time series classification.
- Hybrid approaches: These approaches involve combining algorithms from two or more of the previously mentioned approaches into a single classifier.

State-of-the-art (SOTA) time series classification algorithms achieve high accuracy but often face challenges like slow computation and limited interpretability, making their predictions slow to compute or hard to understand. In some cases, both issues may occur simultaneously. Thus, to tackle these limitations, our work focuses on shapelet-based approaches, which are inherently interpretable, while aiming to improve their efficiency without compromising accuracy.

The Scalable and Accurate Subsequence Transform (SAST) [8] method was proposed as a method builded upon Shapelet Transform (STC) [7]. SAST reduces the computation time of STC by selecting only a few instances per class from a training dataset, resulting in a complexity of $O(nm^3) + O(\text{classifier})$ where n is the number of TS and m is their corresponding length. Although SAST improves computation time compared to STC, its complexity remains cubic with respect to the length of the instances. This can pose significant challenges for certain applications, such as the classification of spectral data, where time series can reach lengths of up to 2844 points [2].

To enhance the scalability of SAST, we propose a novel approach called Random SAST (RSAST). RSAST, based on SAST and STC, eliminates the need to explore every possible set of subseries in a training dataset by using a stratified sampling technique combined with statistical tools, significantly reducing the search space for shapelets, or subsequence candidates that represent a class. As a result, our proposed method substantially reduces computation time while preserving both accuracy and interpretability. RSAST uses fundamental statistical concepts, such as analysis of variance

(ANOVA) [16], the autocorrelation function (ACF) [1], and the partial autocorrelation function (PACF) [1]. Therefore, assuming independence, homoscedasticity, and normality, ANOVA assesses the variability within and between classes, enabling the comparison of mean similarities. Meanwhile, the ACF and PACF, assuming stationarity (a constant behavior of a TS), measure the correlation between observations over time.

Thus, the contributions of our work can be summarized as follows:

- Using a stratified sampling strategy, similar to SAST, but incorporating statistical criteria such as ANOVA, ACF, and PACF to improve the shapelets selection process.
- Providing an open-source implementation of RSAST, based on the SAST model. This novel implementation significantly reduces training time compared to SAST. For example, when applied to the Earthquakes dataset, which is one of the largest datasets from the UCR archive [2], SAST took 2 hours and 37 minutes for model training with an accuracy of 0.68, whereas RSAST completed the same training process in only 6 minutes and 36 seconds with 0.72 of accuracy.

II. BACKGROUND AND RELATED WORKS

A. Definitions

In this section, we provide an overview of the key concepts used to describe RSAST methodology.

- 1) **Time Series Classification:** It is a task in machine learning that involves predicting the class or category of a time series.
- 2) **Time Series:** A time series (TS) is a sequence of data points indexed in time order, where each data point is associated with a specific timestamp. TS can be used within an instance pair $\{\mathbf{x}, y\}$, where \mathbf{x} is the TS with m observations (x_1, x_2, \dots, x_m) , and y is a discrete class variable with $|c|$ possible values. The list of n cases with associated class labels can be represented as $\mathbf{T} = (\mathbf{X}, \mathbf{y}) = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n))$.
- 3) **Classifier:** It refers to a function or mapping that operates on the input space, such as the times series for TSC, and assigns a probability distribution over the possible class variable values.
- 4) **Shapelets:** These are discriminative subsequences [17], denoted as $\mathbf{s} = \{s_1, \dots, s_l\}$ with length l , that capture local patterns within time series and can effectively differentiate between different classes or categories in a time series dataset.
- 5) **Analysis of variance (ANOVA):** It is a statistical test used to determine if there is a significant difference between the means of two or more groups based on the variation in the data [16]. The simplest type of ANOVA test is one-way ANOVA, which examines the effect of a single categorical independent variable (factor) on a continuous dependent variable. The assumptions of this test are the independence, normality and homoscedasticity of the observations among groups.

- 6) **Autocorrelation Function (ACF):** It is a tool used to measure the similarity between a time series and a delayed version of itself, assuming stationarity and finite variance for the data [1].
- 7) **Partial Autocorrelation Function (PACF):** As in ACF [1], it is a measure of the correlation between a time series and a lagged version of itself, after removing the effects of all the shorter lags .

B. Related Work

The levels of accuracy, complexity, and interpretability can vary in TSC methods mostly in relationship with the category they belong to [12].

For example, one of the techniques that achieve the highest accuracy is a convolutional approach, called ROCKET, which stands for Random Convolutional Kernel Transform [3]. ROCKET generates an extensive assortment of convolutional kernels with randomized parameters. These kernels are then utilized to transform the data through convolution and pooling operations like max pooling (identifying the maximum value) and the proportion of positive values (PPV).

Another accurate method is the Hierarchical Vote Collective of Transformation Ensembles (HIVE-COTE), improved through HIVE-COTE v2.0 (HC2) [11] by changing the methods used in the ensemble. So, only Shapelet Transform (STC) is retained, the temporal dictionary ensemble (TDE) [10] replaces the contract Bag-of-SFA-Symbols (cBOSS) [13], serving as the chosen dictionary classifier. Also, the DrCIF [9] method is introduced to handle the interval and frequency representations. Lastly, HC2 included Arsenal approach, an ensemble of ROCKET classifiers.

Notably, as we mentioned before, we will focus on shapelet approaches given their inherently interpretable characteristic. So, the initial shapelet technique found in the literature emerged with the work of [17]. This procedure introduced embedded shapelets within a decision tree classifier, showcasing the potential of employing shape-based features to enhance classification performance for time series data.

Following this primitive procedure, subsequent shapelet-based studies have been dedicated to two primary aspects: refining the accuracy of the algorithm [17] or addressing the inherent computational complexity associated with shapelet methods.

One improvement in accuracy is proposed with the Shapelet Transform Classifier (STC) [7], which takes a multi-step approach to enhance accuracy in time series classification. Firstly, it conducts an exhaustive search for shapelets within the training data. Then, it transforms the time series into distance-based vectors, enabling the selection of the most suitable shapelets among classes using the information gain criterion. Finally, with the selected shapelets, the algorithm constructs a classifier using the transformed dataset.

Moreover, in Fast Shapelets (FS) [15], it was introduced an enhanced version of the decision tree shapelet approach proposed in the original algorithm [17]. This approach focuses on accelerating shapelet discovery. Instead of performing an

exhaustive search at every node, FS utilizes symbolic aggregate approximation (SAX) to discretize and approximate shapelets. SAX is a technique that transforms time series into strings, reducing their dimensionality using piecewise aggregate approximation (PAA) and discretizing them into bins based on equal probability areas of the normal distribution.

Similarly in Learned Shapelets (LS) [5], it is introduced the use of a heuristic gradient descent shapelet search approach instead of the traditional enumeration strategy. Unlike FS and STC, LS is not constrained to identifying shapelets solely from subseries within the training data.

Further, the Scalable and Accurate Subsequence Transform (SAST) was proposed to improve the complexity of STC by reducing the number of patterns to be assessed [8], and as a consequence, making the model faster to train. To accomplish this, SAST reduces the number of shapelet candidates by selecting randomly k instances per class. Furthermore, it does not select the top best shapelets beforehand, as in STC, instead, all the shapelets candidates from the selected instances are used to transform the data.

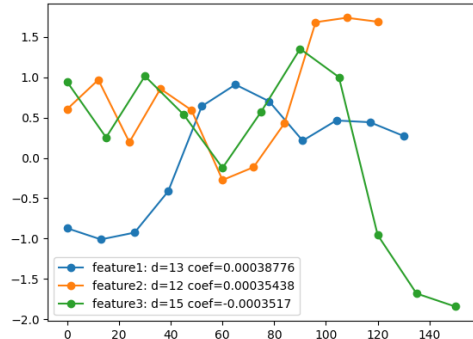
Lastly, a recent advancement for shapelet-based techniques has been introduced with the Random Dilated Shapelet Transform or RDST [6]. This algorithm integrates various convolution techniques. RDST employs dilation with shapelets, which means that some spaces are defined between time points. Hence, a shapelet with dilation d is compared to time points d steps apart when computing distances between the shapelet and time series. The method also uses two features in addition to distance, it uses the position of the minimum distance, and it measures the frequency of occurrences of the shapelet, based on a distance threshold. Consequently, the transformed data comprises 3 times the number of shapelets, which are used at the end of the procedure to train a classifier.

Although, highly accurate, RDST has a deficiency in interpretability since the shapelets found could exhibit dilation, which is not natural for domain experts. An example of the lack of interpretability can be seen on the Coffee dataset from the UCR archive. Here, the first shapelet (see Fig. 1a), according to a feature importance analysis, has a dilation of 13, the second most important shapelet has a dilation of 12 and the third shapelet has a dilation of 15. As a consequence, when these shapelets are plotted on random instances from each class of the training set (see Fig. 1b), it is observed that they do not seem to match perfectly with the time series, as it is the case of other methods like STC, SAST, and our proposed method RSAST (see Fig. 2b). In consequence, it is not clear the pattern that enables the differentiation of classes of the RDST approach.

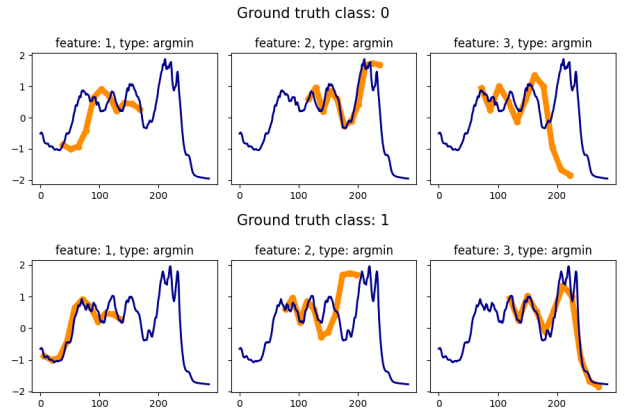
Besides, in RDST, the features are categorized into 3 types: the presence or absence of the shapelet (min), the position (argmin), and the frequency (SO). So, for the interpretation of the method, it is necessary to take into account the feature type as part of the analysis of the results. For instance, in the previous example, the 3 most important shapelets are argmin type for the Coffee dataset (see Fig. 1b), indicating that the location of the shapelets determines their importance. This

result presents shapelets as phase-dependent features, making it difficult for a domain expert to understand its meaning, as the location has to be taken into account. For more details of the RDST method, it is suggested to read [6].

By contrast STC, SAST, and our proposed method RSAST are based on shapelets that are also obtained from the training dataset (see Fig. 2a) but that perfectly match their corresponding class (see Fig. 2b) and does not depend on the location of the shapelet. We will go in deep with this subject in the Section IV-C so as to explain in detail the interpretable characteristic of RSAST.



(a) Top 3 Shapelets Obtained with RDST.

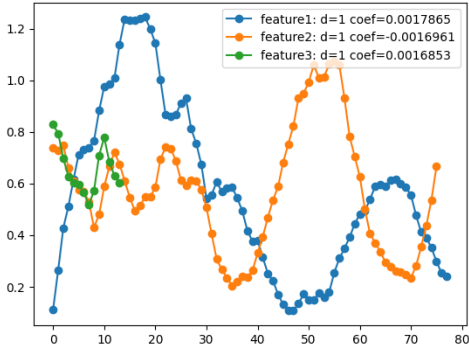


(b) Top 3 shapelets, on TS, learned by RDST.

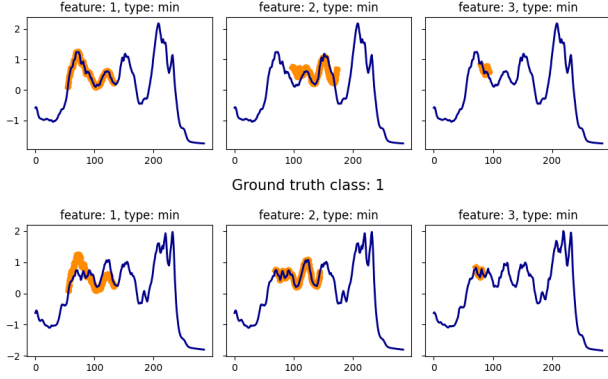
Fig. 1: RDST applied to Coffee dataset.

III. RSAST METHOD

Our method leverages the principle that time series originating from different classes exhibit distinct stochastic properties. Consequently, shapelets, denoted as $(S = \{s_1, \dots, s_l\})$ with length (l) , extracted from (n) time series of length (m) , should be obtained from series (T_i) and (T_j) belonging to different classes $(c_i \neq c_j)$. This selection is guided by the condition that, for a specific time point (t) , there is a significant difference in their means $(|\mu_{i,t} - \mu_{j,t}| > 0)$. Furthermore, to capture and preserve relationships within time series of different classes, for instance, $T_{i,t}$ or $T_{j,t}$ and subsequent points up to $T_{i,t+l}$ or $T_{j,t+l}$, we utilize the significant values of the ACF and PACF. These functions help identify the minimum appropriate lengths (l) of the shapelets that guarantee the preservation of



(a) Top 3 Shapelets Obtained with RSAST
Ground truth class: 0



(b) Top 3 shapelets, on TS, learned by RSAST.

Fig. 2: RSAST applied to Coffee dataset.

these temporal relationships. ACF and PACF are appropriate tools for this in shapelet selection, since, when time series are stationary, meaning their statistical properties such as mean, variance, and autocorrelation remain constant over time, the relationships between points in the ACF and PACF are independent of the phase, and only the lag between them is relevant [1].

A. Main Ideas RSAST

The proposed method starts by calculating a *set of weights* with the set of values along the length m of the time series. An ANOVA test is performed for each of the set of points $T_{\mathbf{w},t}$, where \mathbf{w} represents all the points across all classes at time t , with t ranging from 0 to m . So, this process yields a set of p-values for each $T_{\mathbf{w},t}$ from the ANOVA test, which facilitates the posterior random selection of initial points for shapelets, prioritizing those points along the time series where the differences among classes are most significant.

Then, analogous to the initial SAST algorithm, k , *instances of reference* per class are drawn (without replacement) from the training dataset. So, PACF and ACF are applied to each of these selected instances. The computation of PACF and ACF for each instance reveals a series of highly correlated lagged time points. Here, $\text{PACF}(T_i)$ denotes the direct relationships between $T_{i,t}$ and $T_{i,t+h}$ for a lag h , while $\text{ACF}(T_i)$ includes the indirect effects among these points.

Consequently, ACF and PACF identify the length for related points in the time series for each class (c), avoiding exploring non-correlated possible options as *potential lengths* for the shapelets. Thus, this step eliminates the need to exhaustively explore the entire range of lengths for shapelets within a time series (where the worst case considers a range from 3 to m).

So, with some *potential lengths* (L_c) determined and utilizing the calculated *list of weights* (W), a pre-defined *number of admissible starting points* (p), for the shapelets, can be sampled (without replacement) from the *instances of reference*. As a result, a collection of relevant shapelets are randomly extracted from the *instances of reference* (See Algorithm 1).

Algorithm 1 GenRandShapeletCandidates

Require: Dataset: D_c , Length list: L_c , Random Points: p ,
Weights: W

- 1: $S \leftarrow \emptyset$
- 2: $n, m \leftarrow \text{size}(D_c)$
- 3: **for** $i \leftarrow 1$ to n **do**
- 4: $rPoints \leftarrow []$
- 5: **for** l in L_c **do**
- 6: $rPoints \leftarrow \text{RandSelectPoints}(T_i, W, p)$
- 7: **for** j in $rPoints$ **do**
- 8: $S \leftarrow S \cup \text{SelectSubsequence}(T_i, j, l)$
- 9: **end for**
- 10: **end for**
- 11: **end for**
- 12: **return** S

Lastly, the algorithm involves computing distances between the shapelets and the time series (TS), both of which are z-normalized. This process, similar to SAST and STC, implies finding the minimum distance between each shapelet and all segments of the TS (See line 8 to 14 in Algorithm 2). This process produces a transformed dataset that represents the distances between the shapelets and the TS. So, a classifier is trained using this new data representation (See line 15 in Algorithm 2). A summary of the different steps involved in RSAST is presented in Figure 3.

Posteriorly to the training of the model, a post-hoc analysis can be done to interpret the results [14]. For instance, in the case of a linear model used as the classifier for RSAST, the absolute value of a weight indicate the relative importance of each feature or shapelet for the classification. Therefore, the highest absolute values of the linear model correspond to the most important shapelet for the classification. Figure 2b demonstrates the interpretability of RSAST using the Coffee dataset. The figure shows the three most important shapelets identified by the method.

IV. EXPERIMENTS

RSAST was implemented in Python and developed within the SAST framework, similarly, following the scikit-learn design principles (https://anonymous.4open.science/t/random_sast). The repository also includes the results of experiments

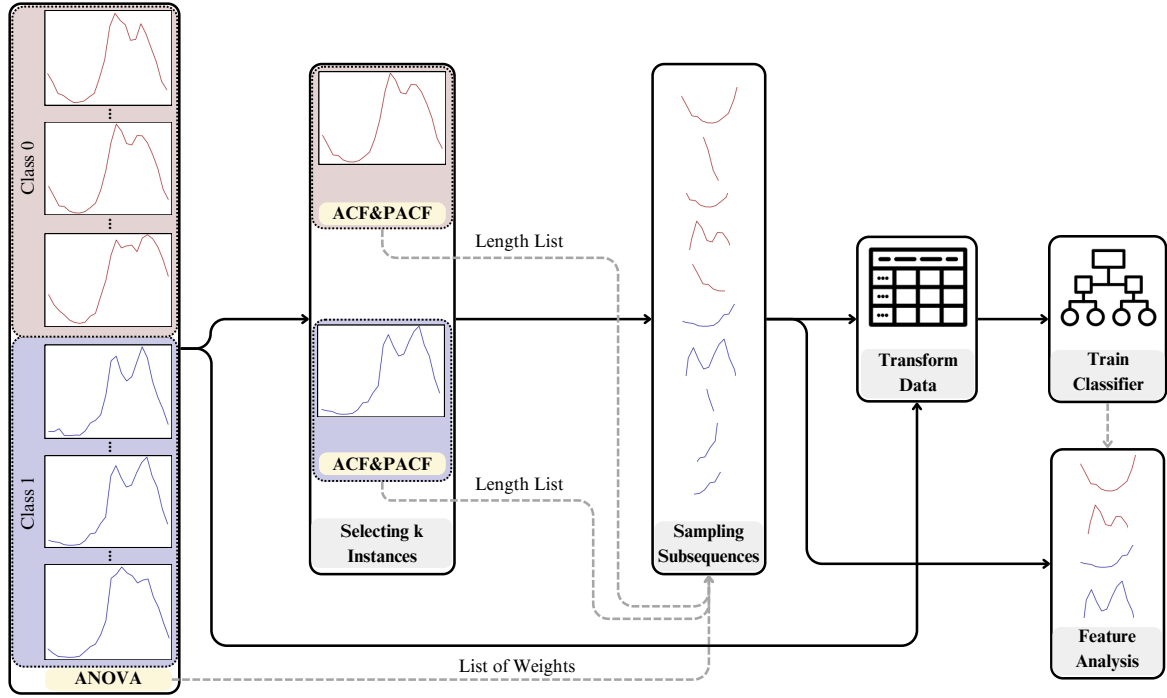


Fig. 3: Overview of RSAST method.

Algorithm 2 Random SAST

Require: Dataset: D , Instances per Class: k , classifier to use:

C , Number of Starting Point: p

1: $n, m \leftarrow size(D)$ \triangleright The size of the dataset is extracted.

2: $W \leftarrow CalculateWeights(D)$

3: $D_c \leftarrow RandSelectInstancesPerClass(D, k)$

4: $L_c \leftarrow GenLenghtListACF(D_c)$

5: $L_c \leftarrow L_c \cup GenLenghtListPACF(D_c)$

6: $S \leftarrow GenRandShapeletCandidates(D_c, L_c, p, W)$

7: $D_f \leftarrow \emptyset$ \triangleright The new transformed dataset is instantiated.

8: **for** $i \leftarrow 1$ to n **do**

9: $x_i \leftarrow []$

10: **for** $j \leftarrow 1$ to $|S|$ **do**

11: $x_i[j] \leftarrow dist(T_i, S_j)$

12: **end for**

13: $D_f \leftarrow D_f \cup \{(x_i, c_i)\}$ \triangleright Each distance and it class are kept.

14: **end for**

15: $clf \leftarrow TrainClassifier(C, D_f)$ \triangleright A classifier is trained on the transformed dataset.

16: **return** (clf, S)

for hyper-parameter tuning, as well as comparisons of RSAST with other shapelet-based approaches and SOTA techniques.

A. Accuracy

In the following section, the results are presented in terms of accuracy, with a comparative analysis of RSAST against the original SAST algorithm and STC. Also, a comparison

is provided between RSAST and other shapelet-based techniques, such as Fast Shapelet (FS) [15], Learned Shapelet (LS) [5], and Random Dilated Shapelet Transform (RDST) [6]. These experiments were conducted on an AMD EPYC 7763 64-Core Processor with 256 CPUs. The supervised classifier employed within RSAST is the Ridge classifier with leave-one-out (LOO) cross-validation for scenarios where the transformed dataset contains more features than instances. In cases where the opposite holds true, a Logistic Regression classifier is utilized. All classifier parameters are left at their default values and are not fine-tuned.

In addition, we employed the Wilcoxon significance test with a significance level of 0.05 to make the comparison of accuracy among the methods. So in order to visualize this, a critical difference diagrams were constructed following the methodology proposed in [4]. These plots show the average ranks across all datasets and represent them on a line, grouping classifiers into cliques (indicated by black lines) where no statistically significant difference in rank exists.

1) *Comparison RSAST, SAST and STC*:: We present the performance of RSAST in terms of accuracy compared to the original SAST algorithm, as well as STC and 1-Nearest Neighbor with Dynamic Time Warping (1NN-DTW) used as a benchmark method. RSAST was evaluated on 128 univariate datasets from the UCR archive [2] using the default train/test splits for 5 independent runs. For SAST, we used the 72 results reported in the original paper [8], that correspond to datasets with default train/test splits from the UCR archive. Similarly, the STC results comes from the 85 Bake Off datasets from the UEA & UCR webpage, using a default train/test splits.

To ensure a correct comparison across the retrieved results for all methods, we restricted the analysis to 56 datasets from the 128 univariate cases of the UCR archive for which results were available for all compared methods. Figure 4, provides a visualization of the relative ranks of each method. Thus, it is observed that RSAST outperforms the accuracy of the TSC benchmark 1NN-DTW. Also, as it is showed by the clique in the critical difference diagram, RSAST achieves a similar level of accuracy to SAST and STC. Thus, RSAST attains comparable accuracy to its baseline counterparts. The one-to-one accuracy comparison of RSAST against SAST, STC, and the benchmark 1NN-DTW is presented in Figure 5. For example, in Figure 5b, RSAST outperformed SAST on 24 datasets, was outperformed on 21 datasets, and tied 11 times.

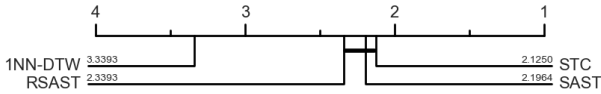


Fig. 4: Critical difference diagram between RSAST, SAST, STC and the benchmark 1NN-DTW.

2) *Comparison RSAST and Shapelet Methods*:: The comparison of shapelet methods includes Fast Shapelet (FS), Learned Shapelet (LS), Random Dilated Shapelet Transform (RDST), and the benchmark method 1NN-DTW. We included the comparative shapelet methods based on the availability of results from the UCR Repository or by locally testing well-established implementations. Thus, RSAST and RDST were evaluated on 128 univariate datasets from the UCR archive using 10 resamples. For the other methods, the results was conducted using the 85 Bake Off datasets [2], also with 10 resamples. Thus, the comparative analysis of shapelet techniques was narrowed to the 85 Bake Off datasets from the UCR archive. Figure 6 illustrates that RSAST outperforms FS and 1NN-DTW. Additionally, RSAST demonstrates a comparable level of accuracy to LS. However, when comparing RSAST and RDST, the latter method achieves better performance on several datasets by leveraging dilated shapelets derived from the training data.

The individual accuracy of RSAST against the other shapelet approaches is summarized in Figure 7. For some datasets the results differ substantially. For instance, in ElectricDevices and Lightning7, RSAST surpasses FS, with scores of 0.82 compared to 0.29, and 0.75 compared to 0.15, respectively. Moreover, RSAST outperforms LS significantly in the OliveOil dataset, achieving a score of 0.87 compared to 0.17. However, RSAST shows notably lower performance in the ShapeletSim dataset, compared to the rest of shapelet methods recording an accuracy of 0.48 compared to 1.0, 0.93, and 0.99 for FS, LS, and RDST, respectively.

B. Scalability

The scalability assessment of RSAST followed a methodology similar to that of SAST [8], so we compared the training

time of these models with respect the time series length and the number of time series within the dataset. For RSAST, we evaluated the method using different number of instances (k), chosen 1 or 10, while the number of random points (p) varied between 10 or 30. For SAST, we set the hyperparameters as recommended by the author. Additionally, we included other scalable methods, such as ROCKET and RDST, in the comparison. The experiments were performed using RSAST integrated with logistic regression for the section focused on increasing the training set size. Further, in relation to the time series length, the experiments were conducted with RSAST in conjunction with a ridge regression classifier. Also, in order to better show the results we re-scale the training time in a logarithmic scale. Both experiments were carried out locally utilizing an Intel Core i5-4300U processor.

1) *Training set size*:: To assess the scalability of the method concerning the training set size, a random oversampling of the Chinatown dataset [2] was performed. Instances were selected with replacement from the original dataset within the range of 2^4 , 2^5 , 2^6 , 2^7 , 2^8 , 2^9 and 2^{10} . This procedure yielded a new dataset with an augmented number of time series. The results are presented in Figure 8 and summarized in Table I, showing the performance of the various tested approaches. Notably, RSAST demonstrates remarkable performance compared to the other approaches. Specifically, RSAST with 1 instance and 10 random points exhibits the best time performance, taking only 0.30s, for 1024 (2^{10}) instances. Additionally, RSAST with 10 instances, and 10 random points took 3.84s for the same number of time series. It is noteworthy that ROCKET is the method that shows the most significant increase in total training time as the number of time series rises, going from 0.97s to 7.79s for 16 and 1024 instances respectively. SAST approach took 3.10s for 1024 instances mirroring the tendency observed with RDST as the number of instances grows. Lastly, RDST achieved a training time of 5.97 seconds for 1024 instances, showing a higher training time than RSAST with 10 instances and 10 random points (3.84s).

Method	# Series	
	16	1024
RSAST: n_random_points=10 nb_inst_per_class=1	0.05s	0.30s
RSAST: n_random_points=10 nb_inst_per_class=10	0.12s	3.84s
RSAST: n_random_points=30 nb_inst_per_class=1	0.05s	0.48s
RDST: n_shapelets=10_000	0.06s	5.97s
ROCKET: num_kernels=10_000	0.97s	7.79s
SAST: min_shapelet_length=3 max_shapelet_length=m	0.14s	3.10s

TABLE I: Summary training time versus training set size (in bold lowest training time in italic highest one).

2) *Time series length*:: To evaluate the scalability of the method in terms of the time series length, the HouseTwenty dataset [2] is utilized, with time series truncated at lengths of 2^5 , 2^6 , 2^7 , 2^8 , 2^9 , and 2^{10} . Figure 9, provides a visual representation of the computation time exhibited by the various methods as the time series length grows. Also, the results are summarized in Table II. Notably, the SAST approach exhibits the highest training computation time, ranging from 0.87s for instances of length 32 to 18 hours, 45 minutes, and 48.02

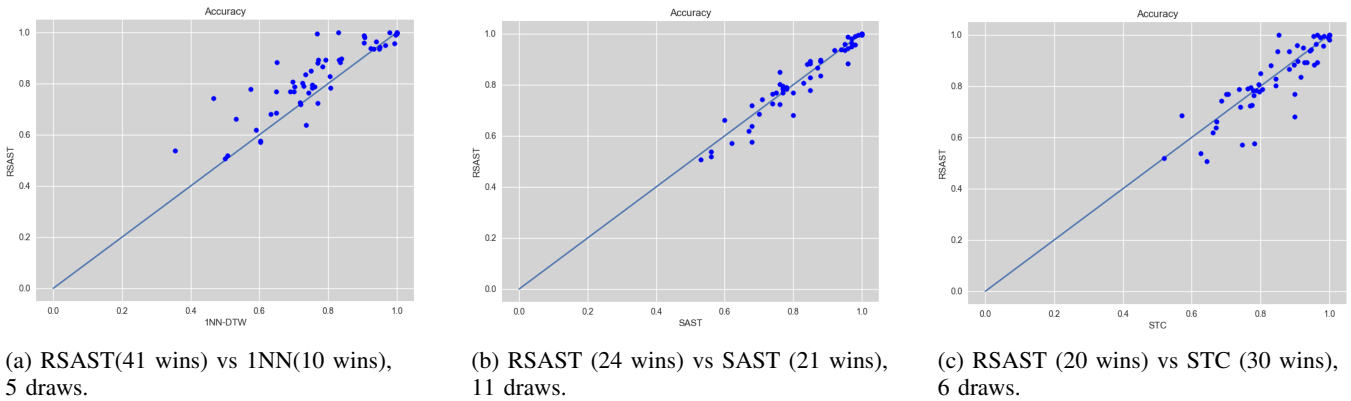


Fig. 5: Pairwise comparison of RSAST against SAST, STC and 1NN-DTW.

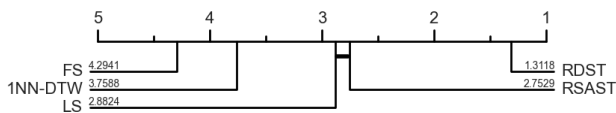


Fig. 6: Critical difference diagram between shapelet approaches and the benchmark 1NN-DTW.

seconds for instances of length 1024 (2^{10}). On the other hand, RSAST with 10 instances and 10 random points, required just 38 minutes and 1.96 seconds for the same length, demonstrating the improved efficiency of RSAST compared to SAST. Additionally, compared to RSAST, RDST and ROCKET tend to be more scalable. However, RSAST follows closely when it is trained with 1 instance and 10 random points. The respective training times for RDST, ROCKET, and RSAST for time series of length 1024 (2^{10}) were 12.45 seconds, 7.96 seconds, and 3 minutes 15.76 seconds.

Method	Length	
	32	1024
RSAST: n_random_points=10 nb_inst_per_class=1	0.05s	3m 15.76s
RSAST: n_random_points=10 nb_inst_per_class=10	0.24s	38m 1.96s
RSAST: n_random_points=30 nb_inst_per_class=1	0.07s	4m 26.37s
RDST: n_shapelets=10_000	0.04s	12.45s
ROCKET: num_kernels=10_000	0.48s	7.96s
SAST: min_shapelet_length=3 max_shapelet_length=m	0.87s	18h 45m 48.02s

TABLE II: Summary training time versus time series length (in bold lowest training time in italic highest one).

C. Interpretability

Similar to the primitive approach [17], STC and SAST, the predictions made by RSAST, can be interpreted through the identification and visualization of the selected shapelets.

Thus, this is done by the application of feature importance analysis. In this process, each feature is linked to a shapelet candidate extracted from a time series. The shapelet candidates that correspond to the most significant features are regarded as the top-performing shapelets.

When a linear classifier is employed, RSAST utilizes the significance of each feature based on the absolute value of

its learned weight. For example, the 3 most relevant shapelets extracted from the Coffee dataset in the UCR archive (see Figure 2a) ranged in length from a short subsequence of 13 points (feature 3) to a longer one of 78 points (feature 1).

Further, these subsequences can also be plotted on their corresponding reference time series, as illustrated for the Coffee dataset in Figure 2b. The first row of the figure displays reference time series from class 0, while the bottom row displays reference time series from class 1. A perfect match between a shapelet (orange line) and a reference time series (blue line) indicates that the shapelet was extracted from that specific series. This visualization provides insights into the discriminatory patterns learned by RSAST to distinguish between the two classes in the dataset.

V. CONCLUSION

In summary, shapelet-based methods are interpretable approaches for time series classification where the aim is to identify distinctive patterns within time series to assign a class to new data. SAST is a state-of-the-art technique that uses these representative subsequences to train a classifier. However, despite its accuracy, SAST faces scalability challenges, particularly with longer time series. To address this, we introduce Random SAST (RSAST). This method improves the computational efficiency of SAST and STC by randomly finding shapelets based on statistical criteria, effectively narrowing the search space of shapelets. RSAST also maintains interpretability, as it relies on extracting continuous subsequences from the training time series, allowing for a clearer understanding of the decision-making process of the model. Nonetheless, RSAST has room for improvement. Notably, its complexity can be further improved with respect to the length of the time series. Additionally, the statistical tests that RSAST employs rely on specific assumptions about the data, including stationarity, independence, normality, and homoscedasticity, which must be considered.

REFERENCES

- [1] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2002.

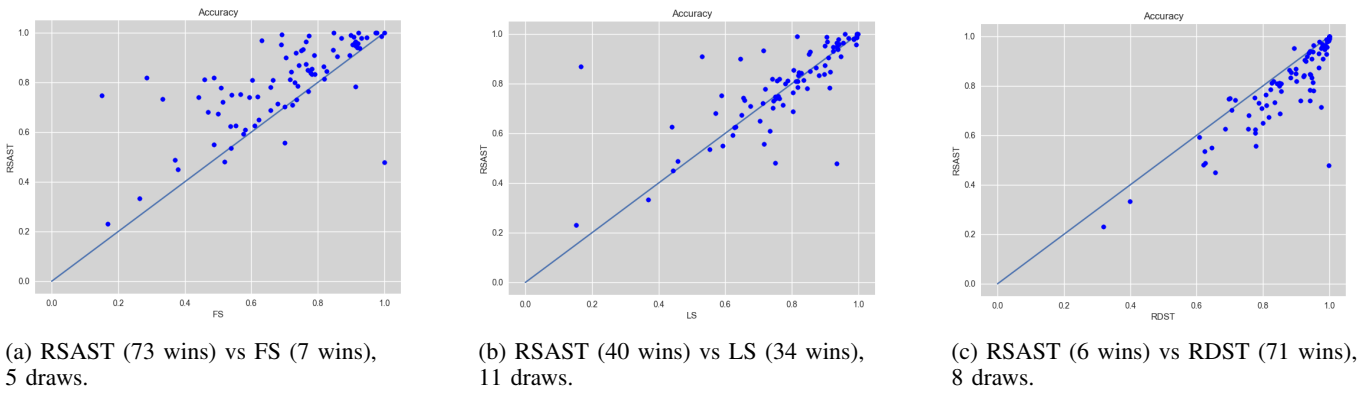


Fig. 7: Pairwise comparison of shapelet approaches.

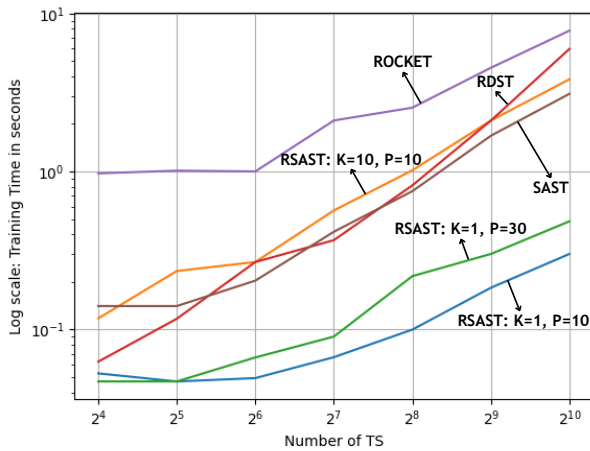


Fig. 8: Training time versus training set size.

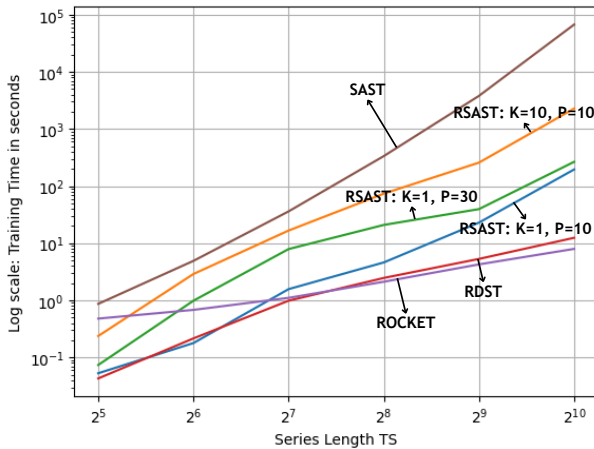


Fig. 9: Training time versus time series length.

- [2] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. [Online; accessed August 18, 2023].
- [3] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using ran-

- dom convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.
- [4] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
- [5] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 392–401, 2014.
- [6] Antoine Guillaume, Christel Vrain, and Wael Elloumi. Random dilated shapelet transform: A new approach for time series shapelets. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 653–664. Springer, 2022.
- [7] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.
- [8] Michael Franklin Mbouopda and Engelbert Mephu Nguifo. Scalable and accurate subsequence transform for time series classification. *Pattern Recognition*, 147:110121, 2024.
- [9] Matthew Middlehurst, James Large, and Anthony Bagnall. The canonical interval forest (cif) classifier for time series classification. In *IEEE International Conference on Big Data*, pages 188–195. IEEE, 2020.
- [10] Matthew Middlehurst, James Large, Gavin Cawley, and Anthony Bagnall. The temporal dictionary ensemble (tde) classifier for time series classification. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 660–676. Springer, 2021.
- [11] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11-12):3211–3243, 2021.
- [12] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, pages 1–74, 2024.
- [13] Matthew Middlehurst, William Vickers, and Anthony Bagnall. Scalable dictionary classifiers for time series classification. In *Intelligent Data Engineering and Automated Learning*, pages 11–19. Springer, 2019.
- [14] Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2020.
- [15] Thanawin Rakthanmanon and Eamonn Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SIAM International Conference on Data Mining*, pages 668–676. SIAM, 2013.
- [16] Lars Sthle and Svante Wold. Analysis of variance (anova). *Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272, 1989.
- [17] Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 947–956. Association for Computing Machinery, 2009.