



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-04311309v2

HAL Id: hal-04311309

<https://hal.science/hal-04311309v2>

Preprint submitted on 2 Dec 2024

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.

RSAST: Sampling Shapelets for Interpretable Time Series Classification

Rojas Varela Nicolas¹, Franklin Mbouopda Michael², and Mephu Nguifo Engelbert¹

¹ LIMOS, University Clermont Auvergne, Clermont–Ferrand, France { nicolas.rojas_varela@doctorant.uca.fr, engelbert.mephu_nguifo@uca.fr }

² Yipa Technologies, Toulouse, France {mf.mbouopda@gmail.com}

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. We evaluated the scalability of RSAST, demonstrating the improvements in computation time compared to its baseline and other SOTA methods. Furthermore, experiments conducted on 128 datasets from the UCR archive showed that RSAST achieves accuracy comparable to the baselines, along with competitive performance against other shapelet-based techniques.

Keywords: Time Series Classification · Shapelet · Scalability · Interpretability · ANOVA · Autocorrelation Function (ACF) · Partial Autocorrelation Function (PACF).

1 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 [1]:

- 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 hard to understand. In some cases, both issues may occur simultaneously. Thus, to overcome 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) [2] method was proposed as a method based on shapelets and builded upon STC [3]. SAST reduces the computation time of STC by selecting only a few instances per class from the 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 [4].

To enhance the scalability of SAST, we propose a novel approach called Random SAST (RSAST). RSAST 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. As a result, our proposed method substantially reduces computation time while preserving both accuracy and interpretability. RSAST is based primarily on fundamental statistical concepts, including analysis of variance (ANOVA) [5], the autocorrelation function (ACF) [6], and the partial autocorrelation function (PACF) [6]. Notably, assuming independence, homoscedasticity, and normality, ANOVA assesses the variability within and between classes, enabling the comparison of mean similarities. Meanwhile, the ACF and PACF are tools in time series analysis that measure the correlation between observations over time, assuming stationarity or a constant behavior of a TS.

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 enhance 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 [4], 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.

2 Related works

There are several SOTA methods that focus on combining different mechanisms to achieve the highest accuracy in the TSC task [1]. For instance, one accurate method is an ensemble approach known as the Hierarchical Vote Collective of Transformation Ensembles (HIVE-COTE) [7], which combines deep learning-based, shapelet-based, dictionary-based and interval-based approaches to achieve high accuracy. Similarly, the Time Series Combination of Heterogeneous and Integrated Embedding Forest (TS-CHIEF) [8] is an ensemble of trees that embed distance-based, dictionary-based, and spectral-based features. Additionally, there are other accurate techniques based on convolutional approaches, such as the Random Convolutional Kernel Transform (ROCKET) [9] and its variants, which mainly aim to create a large assortment of convolutional kernels with randomized parameters in order to transform the original data through convolution and pooling operations. While highly effective, these techniques lack straightforward mechanisms for interpreting their outcomes. In contrast, strictly shapelet-based approaches provide greater interpretability, making them particularly relevant to this work.

2.1 Shapelet Approaches

Shapelets methods are techniques that focus on finding representative subseries that allows distinguishing the different classes of time series. The initial shapelet approach in the literature emerged with the work by [10]. 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 the work by [10], subsequent studies have been dedicated to two primary aspects: enhancing the accuracy of the original shapelet algorithm and addressing the inherent computational complexity associated with shapelet methods.

An improvement in accuracy is proposed through the Shapelet Transform (STC)[11]. STC adopts a multi-step approach to enhance accuracy in time series classification. First, an exhaustive search is performed to identify shapelets,

which are ranked based on their relevance across classes in the training data using the information gain criterion. Next, the time series are transformed into distance-based feature vectors derived from the identified shapelets. Finally, a classifier is trained on the transformed data.

Moreover, Fast Shapelets (FS) [12] introduced an accelerated shapelet discovery technique. Instead of performing an exhaustive search at every node of the decision tree for TS in the original space, as described in [10], FS uses Symbolic Aggregate Approximation (SAX) to discretize and approximate shapelets. SAX 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, reducing the complexity of [10].

Further, in Learned Shapelets (LS) [13], 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.

Additionally, the Scalable and Accurate Subsequence Transform (SAST)[2] aims to improve the complexity of STC by reducing the number of patterns to be assessed, 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. Unlike STC, SAST does not preselect the top-ranked shapelets; instead, it uses all shapelets to transform the data.

Lastly, a recent approach has been introduced with the Random Dilated Shapelet Transform (RDST) [14]. RDST employs dilation with shapelets, which implies 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 shapelet and time series. The method also uses two features in addition to minimum distance, it uses the position of the minimum distance, and the frequency of occurrences of the shapelet, based on a distance threshold. Consequently, the transformed data comprises $3k$ features (for k shapelets) subsequently employed in a classifier.

Although highly accurate, interpreting the results of RDST becomes challenging when the shapelets exhibit dilation, as this is not intuitive for domain experts. An example of this lack of interpretability in RDST can be observed when training the model with the Coffee dataset from the UCR archive (see Fig. 1) [4]. Here, the first shapelet, according to a feature importance analysis [15], has a dilation of 13, the second most important shapelet has a dilation of 12 and the third shapelet has a dilation of 15 (see Fig. 1a). 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. In consequence, it is not clear the distinct pattern which enables the differentiation of classes of the RDST approach.

Besides, in RDST, the features in the transformed dataset have distinct interpretations: they can indicate the presence or absence of the shapelet (min), its position (argmin), or its frequency (SO). Therefore, interpreting the method re-

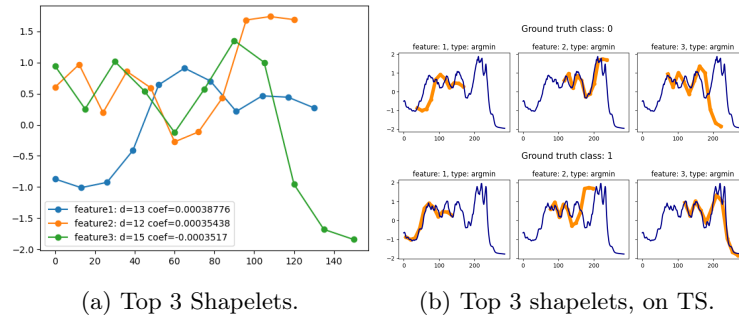


Fig. 1: RDST applied to Coffee dataset.

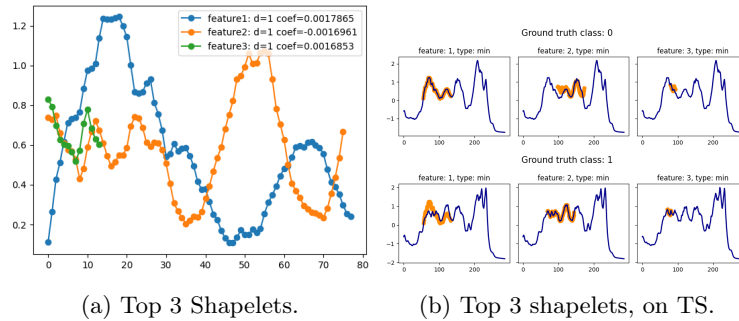


Fig. 2: RSAST applied to Coffee dataset.

quires considering the meaning of the feature as part of the result analysis, which can complicate the interpretation of the outcomes. For instance, in the Coffee dataset (see Fig. 1b), the three most important shapelets are of the argmin type, indicating that their importance is determined by the location of the shapelets rather than their merely presence in the class. For more details of RDST method it is suggested to read [14].

In contrast, STC, SAST, and our proposed method, RSAST, utilize shapelets derived from the training dataset that are straightforward to interpret, as these are continuous series and are phase-independent. An example of RSAST applied to the Coffee dataset is shown in Fig. 2, where the shapelets are better aligned with their corresponding classes, and the features correspond to the presence of the shapelet in the class.

3 RSAST Method

Our method leverages the idea that time series 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 ($c_i \neq c_j$) where there is a significant difference in their means $|\mu_{i,t} - \mu_{j,t}| > 0$ for a specific time point (t). Also, in order to ensure that the initial points for shapelets in $T_{i,t}$ or $T_{j,t}$ are correlated with subsequent points up to $T_{i,t+l}$ or $T_{j,t+l}$, significant values of the ACF and PACF are used to identify the set of lagged points in the time series that are mutually correlated and represent appropriate choices for the length l of the shapelets. This is done because, 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 [6].

3.1 Main Ideas RSAST

Prior to the random selection of shapelets, a *list of weights* is computed with a set of points along the length m of the time series. An ANOVA test is performed on the set of values $T_{\mathbf{w},t}$, where \mathbf{w} represents all values 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 $\text{ACF}(T_i)$ also includes the indirect effects among these points.

Consequently, ACF and PACF identifies related points in the time series for each class, avoiding the consideration of non-correlated lengths as *potential lengths* for the shapelets. Thus, this step eliminates the need to exhaustively explore the entire range of possible lengths for shapelets within a time series (often considering from 3 to m).

So, with the *potential lengths* for the shapelets determined and utilizing the previously calculated *list of weights*, a pre-defined *number of admissible starting points* (p) 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*.

Lastly, the algorithm involves computing distances between the shapelets and the TS, as in SAST and STC, by calculating the minimum distance between each shapelet and all segments of the TS. 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. A summary of the different steps involved in RSAST is presented in Figure 3.

Posteriorly to the training of the model, a post-hoc method can be applied to interpret the results [15]. For instance, in the case of a linear model used as the classifier, the absolute value of a weight indicates 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 2 demonstrates the interpretability of RSAST, as previously introduced, using the Coffee dataset. The figure shows the three most important shapelets identified by RSAST.

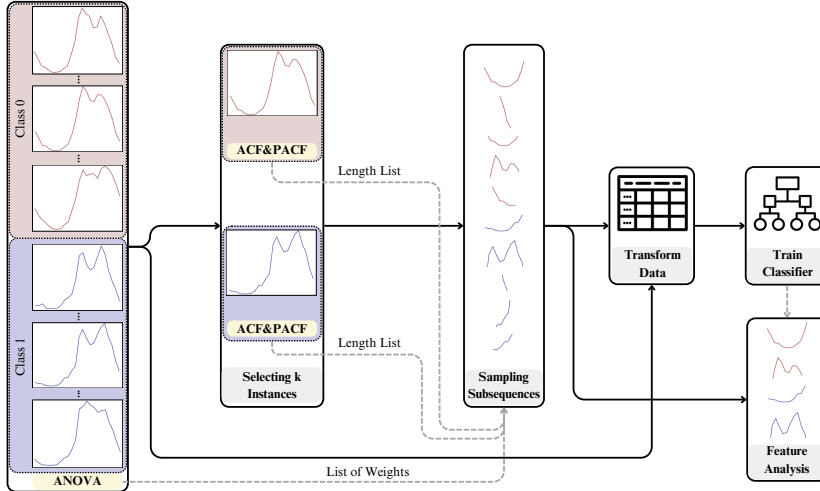


Fig. 3: Overview of RSAST method.

4 Experiments

RSAST was implemented in Python and developed within the SAST framework, following the scikit-learn design principles (https://github.com/nirojasva/random_sast). The repository also includes the results of experiments for hyperparameter tuning, as well as comparisons of RSAST with other shapelet-based approaches and SOTA techniques.

4.1 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) [12], Learned Shapelet (LS) [13], and Random Dilated Shapelet Transform (RDST) [14]. 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 [16]. 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.

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), which served as a benchmark method. RSAST was evaluated on 128 univariate datasets from the UCR archive [4] using the default train/test splits. For SAST, we used the 72 results reported in the original paper [2], 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 fair comparison across methods, we restricted the analysis to 56 datasets from the 128 univariate cases of the UCR archive. Fig.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.

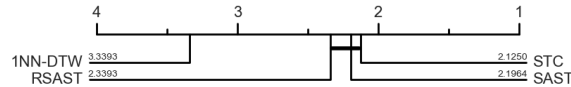


Fig. 4: Critical difference diagram between RSAST, SAST, STC and the benchmark 1NN-DTW

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 include the 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 re-samples. For the other methods, the comparison was conducted using the 85

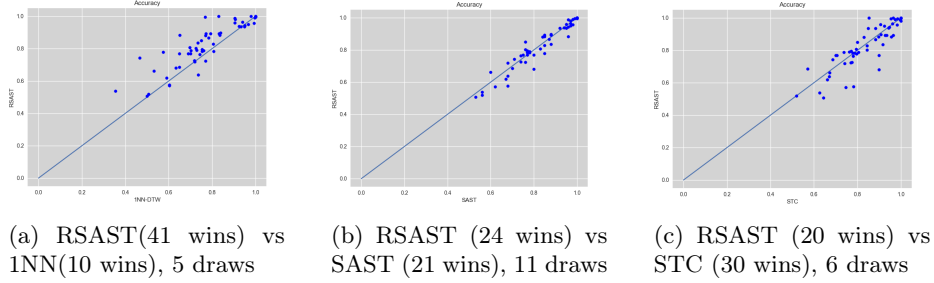


Fig. 5: Pairwise comparison of RSAST against SAST, STC and 1NN-DTW.

Bake Off datasets [4], also with 10 resamples. Consequently, the comparative analysis of shapelet techniques considered 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.

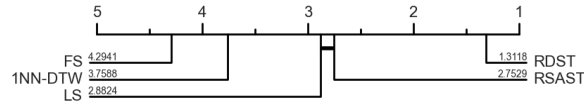


Fig. 6: Critical difference diagram between shapelet approaches and the benchmark 1NN-DTW.

The individual accuracy of RSAST against the other shapelet approaches is summarized in Fig 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.

4.2 Scalability

The scalability assessment of RSAST followed a methodology similar to that of SAST [2], 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,

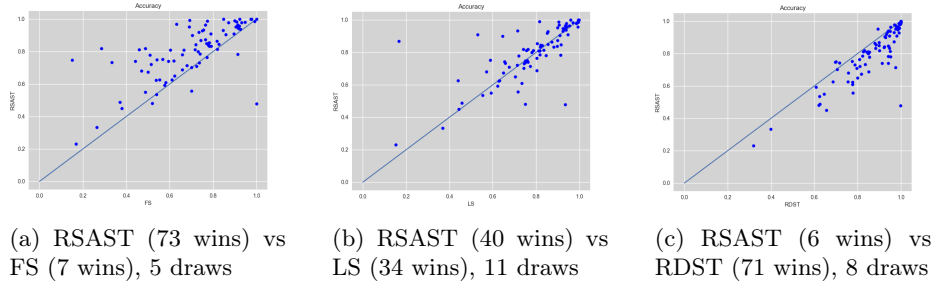
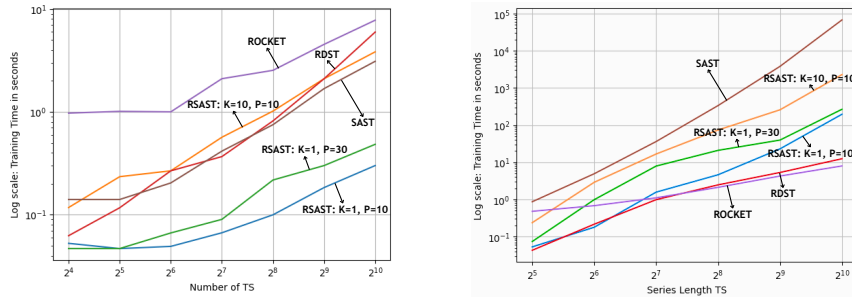


Fig. 7: Pairwise comparison of shapelet approaches.

we evaluated the method using different number of instances (k), ranging from 1 to 10, while the number of random points (p) varied between 10 and 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 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.

Training set size: To assess the scalability of the method concerning the training set size, a random oversampling of the Chinatown dataset [4] 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 8a, 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 0.3s, 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 experiences 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. The SAST approach took 3.1s 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 similarity to RSAST with 10 instances and 10 random points.

Time series length: To evaluate the scalability of the method in terms of the time series length, the HouseTwenty dataset [4] is utilized, with time series truncated at lengths of 2^5 , 2^6 , 2^7 , 2^8 , 2^9 , and 2^{10} . Figure 8b, provides a visual rep-



(a) Training time versus training set size (b) Training time versus time series length

Fig. 8: Scalability RSAST

resentation of the escalating computation time exhibited by the various methods as the time series length grows. Notably, the SAST approach exhibits the highest training computation time, ranging from 0.87 seconds for instances of length 32 to 18 hours, 45 minutes, and 48.02 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, when comparing RSAST with RDST and ROCKET, these methods 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.

5 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), a method that improves computational efficiency by randomly finding shapelets based on statistical criteria, effectively narrowing the search space and enhancing overall efficiency. RSAST also maintains interpretability, as it relies on extracting continuous subsequences from the training time series, allowing for a clearer understanding of the model’s decision-making process. Nonetheless, RSAST has room for improvement. Notably, its complexity can be further improved with respect the length of the time series, and also 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. M. Middlehurst, P. Schäfer, and A. Bagnall, “Bake off redux: a review and experimental evaluation of recent time series classification algorithms,” *Data Mining and Knowledge Discovery*, pp. 1–74, 2024.
2. M. F. Mbouopda and E. Mephu Nguifo, “Scalable and accurate subsequence transform for time series classification,” *Pattern Recognition*, vol. 147, p. 110121, 2024.
3. J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, “Classification of time series by shapelet transformation,” *Data Min. Knowl. Discov.*, vol. 28, no. 4, p. 851–881, 2014.
4. H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, “The ucr time series archive,” *CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019, [Online; accessed August 18, 2023].
5. L. Sthle and S. Wold, “Analysis of variance (anova),” *Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 4, pp. 259–272, 1989.
6. P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. Springer, 2002.
7. M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, “Hivecote 2.0: a new meta ensemble for time series classification,” *Machine Learning*, vol. 110, no. 11-12, pp. 3211–3243, 2021.
8. A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, “Ts-chief: a scalable and accurate forest algorithm for time series classification,” *Data Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 742–775, 2020.
9. A. Dempster, F. Petitjean, and G. I. Webb, “Rocket: exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.
10. L. Ye and E. Keogh, “Time series shapelets: A new primitive for data mining,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2009, p. 947–956.
11. A. Bostrom and A. Bagnall, “Binary shapelet transform for multiclass time series classification,” *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII: Special Issue on Big Data Analytics and Knowledge Discovery*, pp. 24–46, 2017.
12. T. Rakthanmanon and E. Keogh, “Fast shapelets: A scalable algorithm for discovering time series shapelets,” in *proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 668–676.
13. J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, “Learning time-series shapelets,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 392–401.
14. A. Guillaume, C. Vrain, and W. Elloumi, “Random dilated shapelet transform: A new approach for time series shapelets,” in *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, 2022, pp. 653–664.
15. C. Molnar, *Interpretable Machine Learning*. Leanpub, 2020. [Online]. Available: <https://books.google.fr/books?id=jBm3DwAAQBAJ>
16. J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.