Extracting Features from Random Subseries: A Hybrid Pipeline for Time Series Classification and Extrinsic Regression

Matthew Middlehurst and Anthony Bagnall

School of Computing Sciences, University of East Anglia, UK.
m.middlehurst@uea.ac.uk, ajb@uea.ac.uk

Abstract. In time series classification (TSC) literature, approaches which incorporate multiple feature extraction domains such as HIVE-COTE and TS-CHIEF have generally shown to perform better than single domain approaches in situations where no expert knowledge is available for the data. Time series extrinsic regression (TSER) has seen very little activity compared to TSC, but the provision of benchmark datasets for regression by researchers at Monash University and the University of East Anglia provide an opportunity to see if this insight gleaned from TSC literature applies to regression data. We show that extracting random shapelets and intervals from different series representations and concatenating the output as part of a feature extraction pipeline significantly outperforms the single domain approaches for both classification and regression. In addition to our main contribution, we provide results for shapelet based algorithms on the regression archive datasets using the RDST transform, and show that current interval based approaches such as DrCIF can find noticeable scalability improvements by adopting the pipeline format.

1 Introduction

Time series classification (TSC) is the task of predicting a categorical target variable from time series data. The field of TSC has received rapid development in recent years, in part due to the continued maintenance and expansion of the University of California, Riverside (UCR) dataset archive for TSC [9]. Time series extrinsic regression (TSER), like more traditional regression tasks for machine learning, has a continuous target variable. Both tasks differ from standard machine learning in that each data attribute takes the form of a series of ordered values, with discriminatory features found in the shape and frequency of patterns within the series.

TSER has not received the same attention in literature as TSC has, and until recently has not had a collection of datasets comparable to the UCR archive to benchmark algorithms with. A collection of 19 datasets were introduced by Tan *et al.* from Monash University [36], recently further expanded to 63 datasets by researchers at the University of East Anglia (UEA) [19]. A few algorithms

proposed for TSC have been adapted for TSER with mixed success. These algorithms are mostly simple adaptations, using an unsupervised transformation in combination with a vector classifier or regressor. On 62 datasets from the expanded TSER archive, only the Fresh Pipeline with Rotation Forest Classifier (FreshPRINCE) [27] and Diverse Representation Canonical Interval Forest (DrCIF) [29] were significantly better than a Rotation Forest (RotF) [32] benchmark using root-mean-square error (RMSE) as a performance metric [19].

For TSC problems the best approach should consider the discriminatory features present in the series, i.e. whether the presence of a pattern or its frequency is discriminatory, or if patterns are phase-dependent or phase-independent. In the absence of expert knowledge, hybrid approaches encompassing multiple feature extraction approaches have shown to perform more accurately than single domain algorithms [1, 29, 30, 35, 11]. We explore whether this improvement through incorporating multiple domains translates to TSER using a simple pipeline of unsupervised transformations from different feature domains. While hybrid algorithms such as the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [23, 29] and Time Series Combination of Heterogeneous and Integrated Embedding Forest (TS-CHIEF) [35] have already shown to perform accurately on the UCR archive compared to single domain algorithms, modifying these algorithms to accept continuous values would be a complex process which would go beyond the simple exploration we wish to present. By using unsupervised transformations, the only change made to the hybrid algorithm between tasks is the base estimator used.

Our hybrid pipeline makes use of two transformations, both of which randomly select subseries to extract features from. The algorithm selects features from the interval feature domain with a transformation based on the DrCIF ensemble, and from the shapelet feature domain using the Random Dilated Shapelet Transform (RDST). Both of these algorithms have shown to perform accurately in their feature group for TSC on the UCR archive [30]. Our pipeline involves transforming the input series into multiple representations such as first-order differences and periodograms, then extracting and concatenating features for a vector classifier using our transformations. We show that the pipeline is significantly more accurate than DrCIF and RDST on 112 UCR datasets, and that it also outperforms both algorithms on 55 regression TSER problems.

We structure the rest of this paper as follows. Section 2 discusses the background and related works. In Section 3 we describe our pipeline in greater detail. Section 4 discussed our experimental methodology and provides details for reproducibility, followed by Section 5 which presents our results on the UCR and TSER archives. In Section 6 we summarise our findings and conclude.

2 Background and Related Work

Both TSC and TSER are tasks where the objective is to create a function which maps input time series data to a target variable using a training set of time series and label pairs. Input case pairs $(\boldsymbol{X}, y)$ hold a time series $\boldsymbol{X}$ containing

d channels $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d\}$ with m real-valued ordered time points $\boldsymbol{x} = \{t_1, t_2, \ldots, t_m\}$ and a target label y. For TSC y is a discrete class label from c possible class values, while for TSER y is a scalar value. Case pairs are grouped into datasets of n pairs $\boldsymbol{T} = \{(\boldsymbol{X}_1, \boldsymbol{y}_1), (\boldsymbol{X}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{X}_n, \boldsymbol{y}_n)\}$. Datasets where the time series contains only a single channel are univariate time series problems, while those with more than a single channel are multivariate. It is not always the case that all time series in a dataset will have the same number of time points, but we restrict this work with the assumption that all series have the same length.

A comparison of TSC algorithms in 2017 [1] created a taxonomy of TSC algorithms based on the types of feature extracted, sorting the algorithms used into different domains. In 2017 there were six categories, and recent updated comparison has increased this to eight categories [30]. In the following we outline these categories, including descriptions of relevant algorithms and those we use in our Section 5 experiments. While we describe all categories in the following for context of the wider field and different approaches for TSC, our main interest in this study lies with interval-based approaches, shapelet-based approaches and hybrid approaches.

Distance-based algorithms make use of distance measures to compare time series, usually using a nearest-neighbour (NN) algorithm to make predictions. A popular benchmark is the elastic distance measure Dynamic Time Warping (DTW) using a 1-NN classifier or regressor. There are many elastic distances for time series proposed, which have been used individually and as part of ensembles. Proximity Forest (PF) [25, 21] is a distance-based ensemble making use of different distance measures in its ensembled trees.

Dictionary-based look for the frequency of recurring patterns as a discriminatory feature. These are most commonly found through converting time series into a sequence of discrete symbolic words, forming a bag-of-words to compare cases. More recent methods run multiple configurations of word extraction techniques to form an ensemble such as the Temporal Dictionary Ensemble (TDE) [26] or as part of a pipeline with feature selection like Word Extraction for Time Series Classification (WEASEL) [33, 34].

Feature-based algorithms are techniques which extract a feature vector of summary statistics to be used as part of a simple pipeline. These pipelines are mainly made up of two components, the transformation to convert the series to features, and a base estimator to build a model and make predictions using said features. An example is FreshPRINCE [27], a pipeline of the TSFresh [8] features and a rotation forest [32] which has performed as well as more complex algorithms from other domains for TSC and is a top performer on the TSER archive [19]. The iFx [17] for TSER extracts many summary statistics from different series representations and subseries as features for a Bayesian method.

Convolution-based approaches make use of many randomly initialised convolution kernels in conjunction with the linear classifier as part of a pipeline. The Random Convolutional Kernel Transform (ROCKET) [10] and its derivatives such as MultiROCKET [37] and Hydra [11] fall under this category. Our

approach shares similarities with the MultiROCKET-Hydra pipeline proposed in [11], which concatenates the features of both transforms for a pipeline.

Deep learning, like other machine learning fields, is a popular topic for time series tasks. The InceptionTime [14] is currently the best performing deep learner for TSC. The version of InceptionTime we use is an ensemble of 5 networks (InceptionE) proposed in the original publication.

Interval-based approaches select phase-dependent subseries from the input series to derive features from. By selecting many subseries the goal is to derive features that may be otherwise be obscured by irrelevant activity in the series should the whole series be used. Most interval base approaches use a random forest approach [13, 15, 28]. The DrCIF algorithm follows this, randomly selecting multiple intervals and subsampling the Catch22 [24] features for each tree. The Randomised Supervised Time Series Forest (R-STSF) [7] breaks this mould of ensemble approaches, using a pipeline approach for its extracted intervals.

Shapelet-based algorithm find phase-independent discriminatory subseries, looking for the presence of a pattern anywhere in the time series rather than its frequency or at specific time points. Shapelet models compare extracted shapelets to series using a function $sDist()$, which finds the shorted distance from the shapelet to all subseries of the same length. The Shapelet Transform Classifier (STC) [22, 4] algorithm is a pipeline algorithm which creates a feature vector of $sDist()$ values using a filtered set of shapelets and a rotation forest classifier. RDST is an algorithm based on the shapelet transform which we cover in more detail in the following section. The Multiple Representations Sequence Mine (MrSQM) [31] follows an approach of discretising series into words using multiple differently parameterised methods and uses the presence of selected subsequences in any part of the full word as features for a logistic regression model.

Hybrid algorithms incorporate two or more of the above categories in a single algorithm with the aim of leveraging the strengths of each domain included. At the time of writing the most accurate hybrid algorithm on the UCR archive is HIVE-COTE v2 [29, 30], a weighed ensemble of high performance algorithms from other domains. The HC2 ensemble includes DrCIF, TDE, STC and an ensemble of ROCKET classifiers called the Arsenal. The Time Series Combination of Heterogeneous and Integrated Embedding Forest (TS-CHIEF) [35] also takes an ensemble approach to combining feature domains, but creates a homogenous forest of trees which extract hybrid features at each node rather than a heterogenous ensemble like HC2.

3 A Randomised Shapelet and Interval Transformation Pipeline

The pipeline classifier and regressor we use in our experiments is a hybrid of interval and shapelet based approaches. For brevity, we refer to this pipeline as the Randomised Interval-Shapelet Transformation (RIST) pipeline going forward. Both of these feature domains extract random subseries from the input

series, but how these subseries are used and the features extracted from them differ.

For the interval half of RIST we draw from the DrCIF [29] algorithm. Instead of extracting a small amount of intervals for a single tree as part of an ensemble, we extract a larger amount of intervals in a singular step to concatenate with the shapelet transform output. For RIST we extract i intervals of random length and size. From these subseries, 30 summary statistics are extracted. These are the Catch22 [24] features used by DrCIF, as well as the mean, standard-deviation, slope, median, interquartile range, min, max, and proportion of positive values. Algorithm 1 describes the interval portion of the transformation.

Algorithm 1 Intervals(A list of n series of length m with d channels, $\boldsymbol{X}$)

Parameters: the number of intervals i

$\boldsymbol{X'} \leftarrow$ initialize matrix of dimensionality $n \times (i30)$
for $j \leftarrow 1$ to i **do**
  $b = rand(1, m-3)$ { *interval position* }
  $l = rand(3, m/2)$ { *interval length* }
  $o = rand(1, d)$ { *interval channel* }
  for $t \leftarrow 1$ to n **do**
    for $f \leftarrow 1$ to 30 **do**
      $\boldsymbol{X'}_{t,(j-1)30+f} \leftarrow summaryStat(f, \boldsymbol{X_{t,o,b:l}})$
$\boldsymbol{X'} \leftarrow pruneIdenticalIntervals(\boldsymbol{X'})$
return $\boldsymbol{X'}$

The shapelet half of RIST leverages the RDST [20] transformation without modifications. RDST randomly selects a large number of random shapelets from the train data. Unlike the original Shapelet Transform (ST) [22] algorithm, RDST does not evaluate shapelets using information gain or any other metric to determine the quality of the shapelet to act as a filter. RDST only prunes any identical shapelets from its initial random selection. The shapelets extracted by RDST use dilation as the primary method of diversifying extracted shapelets rather than shapelet length. Using dilation in subseries is a technique which primarily used in convolution based methods such as ROCKET [10, 37], but has been introduced to other algorithm domains recently [20, 34]. A shapelet with a dilation value of d compares time points which are d steps apart, a d value of 1 will have no gaps between values sampled for the shapelet, while a value 2 will sample every other value.

The standard shapelet distance (sDist) method is applied by RDST. To compare a shapelet to a full time series, a sliding window is run across the series calculating the distance to all subseries of the same length as the shapelet, but with the addition of dilation. As well as taking the minimum distance from all subseries as a feature, RDST also extracts the position of the minimum distance subseries and the number of occurrences of the shapelet determined by a similarity threshold. These additional features incorporate spatial information as well

as pattern occurrence information seen in dictionary based approaches into the extracted features.

When selecting its shapelets, RDST randomly initialises the dilation value of shapelet; whether the shapelet distance is z-normalised; the train case and position in the series the shapelet is extracted from; and the similarity threshold used in the shapelet occurrence feature. For multivariate time series, a two-dimensional shapelet is extracted and used to compare the distance of all channels. A simplified version of the shapelet extraction algorithm is displayed in Algorithm 2. For exact values used when selecting random shapelets, we recommended viewing the original publication or the implementation we direct to in Section 4.

Algorithm 2 Shapelets(A list of n series of length m with d channels, $\boldsymbol{X}$)

Parameters: the number of shapelets s

$\boldsymbol{X'} \leftarrow$ initialize matrix of dimensionality $n \times (s3)$
for $j \leftarrow 1$ to s **do**
  $dil, thr, norm \leftarrow shapeletParams()$ { *randomly select shapelet parameters* }
  $o \leftarrow randint(1, n)$
  $pos \leftarrow randint(1, m - dil10)$ { *randomly select position to extract from* }
  $\boldsymbol{A} \leftarrow dilatedSubseries(\boldsymbol{X_o}, pos, 11, dil)$ { *extract shapelet, always length 11* }
  for $t \leftarrow 1$ to n **do**
    $\boldsymbol{d} \leftarrow sDist(\boldsymbol{A}, \boldsymbol{X_t}, dil, norm)$ { *distances between* $\boldsymbol{A}$ *and all subseries* }
    $\boldsymbol{X'_{t,(j-1)3+1}} \leftarrow min(\boldsymbol{d})$
    $\boldsymbol{X'_{t,(j-1)3+2}} \leftarrow argmin(\boldsymbol{d})$
    $\boldsymbol{X'_{t,(j-1)3+3}} \leftarrow occurrences(\boldsymbol{d}, thr)$
return $\boldsymbol{X'}$

Extracting intervals from different series representations has shown to improve accuracy over just extracting intervals from the base series [6, 7, 29]. For RIST we also extract features from different series representations by applying the series-to-series transformations used in the R-STSF algorithm, which have also seen use in many other published TSC algorithms. These are the first order differences [6, 29, 37], the periodogram of the series [6, 29, 15] and the series autoregression coefficients [7]. We run our shapelet and interval transformations on each of these series representations as well as the base series, then concatenate them for use in a feature vector classification or regression algorithm. The RIST pipeline is described in Algorithm 3.

4 Experimental Methodology and Reproducibility

We run our experiments using two time series dataset archives. Our classification experiments are run using 112 datasets from the UCR time series archive[1] [9]. We

[1] `https://www.timeseriesclassification.com/dataset.php`

Algorithm 3 RIST(A list of n cases of length m with d channels, $\boldsymbol{T} = (\boldsymbol{X}, \boldsymbol{y})$)

Parameters: the number of intervals i, the number of shapelets s, the feature vector estimator est

Let $\boldsymbol{V}$ be a $4 \times n \times d$ matrix of series with variable length, containing the base series, the periodograms, the first order differences and the autoregression coefficients
$\boldsymbol{X'} \leftarrow []$
for $j \leftarrow 1$ to $|\boldsymbol{V}|$ **do**
  $\boldsymbol{I} \leftarrow Intervals(\boldsymbol{V_j}, i)$
  $\boldsymbol{X'} \leftarrow \boldsymbol{X'} + \boldsymbol{I}$ { *concatenate feature vectors* }
  $\boldsymbol{S} \leftarrow Shapelets(\boldsymbol{V_j}, s)$
  $\boldsymbol{X'} \leftarrow \boldsymbol{X'} + \boldsymbol{S}$ { *concatenate feature vectors* }
$est.buildEstimator(\boldsymbol{X'}, \boldsymbol{y})$

exclude all datasets from the archive which contain unequal length series or series with missing values from our selection. All classification datasets used are univariate, containing a single channel time series for each case. For our extrinsic regression experiments, we use 55 datasets out of the 63 total from the TSER repository[2] [36] and datasets from a proposed extension[3] [19]. The NewsHeadlineSentiment; PPGDalia-equal-length; VentilatorPressure; AustraliaRainfall; NewsTitleSentiment; BIDMC32SpO2; BIDMC32HR; and BIDMC32RR datasets are excluded solely due to time constraints. The TSER archive includes both univariate and multivariate datasets, of which we use both to supplement the low volume of univariate datasets. With the inclusion of multivariate TSER datasets it is sensible to ask why the UEA archive of multivariate TSC datasets [2] is not included. We again exclude these due to time constraints in the running of our experiments, but note that many of the algorithms including our proposed one are multivariate capable and these datasets should be explored in future work.

We present the performance of an algorithm on a dataset as an average over 5 resamples. Both UCR and TSER archives provide a default train and test split, which we use for the first resample. The remaining runs are resampled randomly from the provided split in a stratified manner for the UCR datasets, and fully random for the TSER data. Each algorithm and data resample random number generation is seeded using the resample index to help ensure reproducibility.

For comparison of multiple classifiers over multiple datasets, an adaptation of the critical difference diagram [12] is used. The post-hoc Nemenyi test is replaced using pairwise Wilcoxon signed-rank tests using our averaged scores. Cliques are formed using the Holm correction, following recommendations from [16, 3]. We compare our classification algorithms using accuracy, and our regression algorithms using RMSE following [36, 19].

[2] `http://tseregression.org/`
[3] `https://tsml-eval.readthedocs.io/en/latest/publications/2023/tser_archive_expansion/tser_archive_expansion.html`

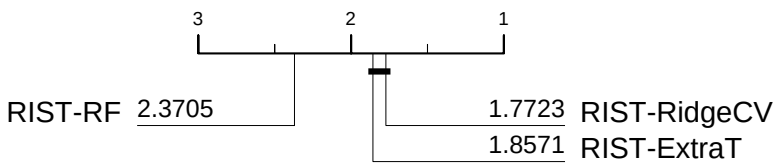

Fig. 1. Accuracy critical difference diagram for RIST with different base classifiers. Displays the average accuracy rank averaged over 5 resamples on 112 UCR datasets.

All the tools to run our experiments are available through the *tsml-eval*[4] package, primarily using implementations from the *aeon*[5] toolkit. More details on reproducing our experiments and results files can be found on the companion webpage[6].

5 Results

In the following, we present summarised results for RIST and relevant algorithms for both archives. For RIST we set the number of intervals extracted to $i = (sqrt(m) * sqrt(d) * 15 + 5)$ and the number of shapelets to $s = (sqrt(m) * 200 + 5)$. Both of these are functions of the dataset series length and number of dimensions, taking into account that the series length may change per series representation.

Prior to our main results, we show results for different base estimators used in RIST, showing that this selection can have a large impact on overall results. The base estimators we compare include a linear Ridge estimator using cross-validation (RidgeCV) which is a commonly used base classifier for TSC [10, 37, 20]. Also compared are a Random Forest (RF) [5] which is a well known and popular baseline, and the Extra Trees (ExtraT) [18] algorithm, another random tree base ensemble used by R-STSF [7].

5.1 112 UCR archive classification datasets

Figure 1 compares the RIST transform using different feature vector classifiers. Both the ridge and extra trees classifiers show no significant difference when used as a base. As the extra trees classifier was quicker at 7 minutes on average to process the UCR datasets against the 10 minutes of the ridge classifier, we use that as our base. Figure 2 shows a pairwise diagram comparing the average accuracy of the extra trees classifier against the ridge classifier for all datasets. Despite using the same seeded transformation, the difference in accuracy between both algorithms can be quite large for some datasets.

[4] `https://github.com/time-series-machine-learning/tsml-eval`
[5] `https://www.aeon-toolkit.org/`
[6] `https://tsml-eval.readthedocs.io/en/latest/publications/2023/rist_pipeline/rist_pipeline.html`

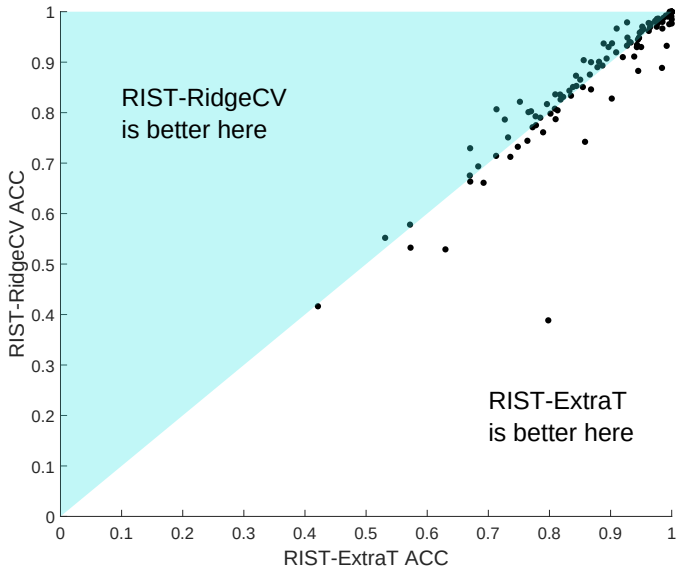

Fig. 2. Scatter plot of RIST using an extra trees and ridge base classifier. Compares the average accuracy over 5 resamples for each of the 112 UCR archive datasets. RIST-RidgeCV Win/Draw/Loss 60/8/44.

We compare RIST against other TSC algorithms in Figure 3. Similar to the RDST and R-STSF pipelines, we include a single domain interval pipeline which just our interval transformer and an extra trees classifier to help gauge the impact of the pipeline structure vs ensemble structure of DrCIF. The simple RIST pipeline concatenating transform outputs significantly outperforms both DrCIF and RDST, the algorithms the transforms are based on. The only algorithm which it performs significantly worse than in our comparison is HC2, another hybrid containing more feature domains and more complex algorithms.

A comparison of runtime against accuracy rank is shown in Figure 4. RIST is not as fast as RDST, R-STSF or ROCKET, but compares favourably to more complex algorithms. While HC2 is significantly more accurate than RIST, it is also close to 40 times slower to build on average. The interval transformation pipeline we included shows no significant difference in performance to DrCIF and is an order of magnitude faster than DrCIF. To achieve similar scalability improvements using the ensemble structure, the amount of DrCIF trees built and intervals extracted would have to be significantly reduced, which is likely to impact performance considerably.

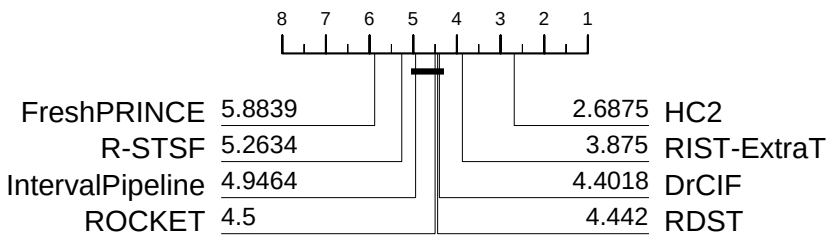

Fig. 3. Accuracy critical difference diagram comparing RIST with seven classification algorithms. Displays the average accuracy rank averaged over 5 resamples on 112 UCR datasets.

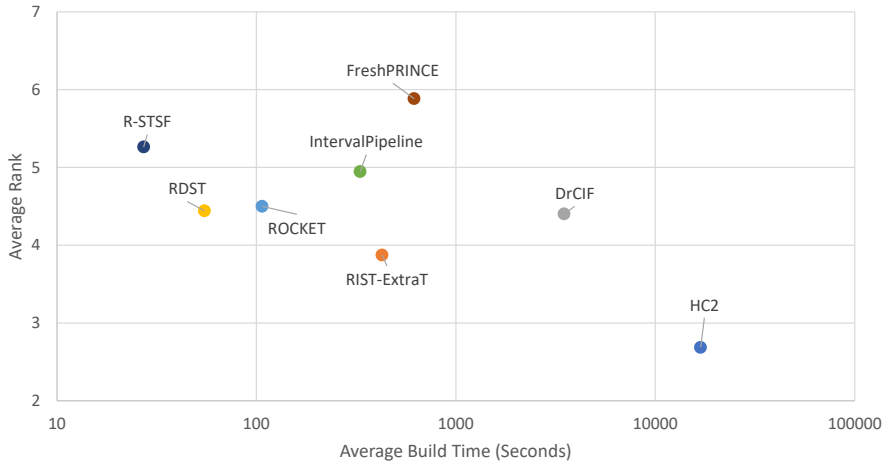

Fig. 4. A comparison of classifiers accuracy rank and build time averaged over 112 UCR problems. The build time is on a log scale.

5.2 55 TSER archive datasets

In our previous experiments we show that a combination of shapelet and interval features from RIST outperforms the single domain classifiers the features are derived from on the UCR archive. We now experiment to see if this is the case for the newly introduced TSER archive as well.

Figure 5 compares different base regressors for RIST as we previously did for classification. While for the classification task the random forest classifier was significantly worse, the random forest regressor has seemingly swapped position with the ridge regressor. In previous comparisons on the TSER archive, ROCKET which also uses a ridge regressor performed below expectations considering its success in classification [19]. It is possible that the selection of base estimator could play a part in this underperformance. Of the two best performing base regressors, the extra trees algorithms is faster to build on the TSER

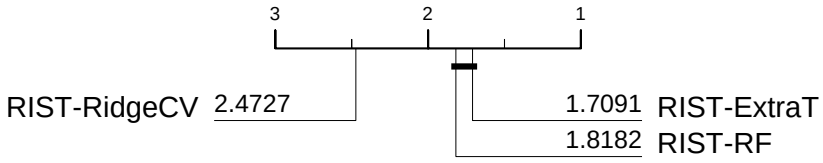

Fig. 5. RMSE critical difference diagram for RIST with different base regressors. Displays the average RMSE rank averaged over 5 resamples on 55 TSER datasets.

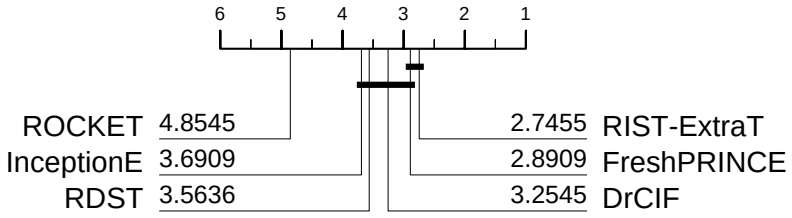

Fig. 6. RMSE critical difference diagram comparing RIST with five regression algorithms. Displays the average RMSE rank averaged over 5 resamples on 55 TSER datasets.

archive. For this reason and to keep consistency with the classification version, we use it as a base for our regression experiments.

A critical difference diagram comparing RIST and competitive TSER algorithms on the 55 datasets is shown in Figure 6. RDST was not included in previous publications experimenting with the TSER archive, and places middle of the pack in a clique with inception time, DrCIF and FreshPRINCE. RIST shows no significant difference to FreshPRINCE, but once again performs significantly better than both DrCIF and RDST. While there are other factors at play which could contribute to this increased performance, we believe it is likely that the same assumption regarding the performance of hybrid approaches in TSC also applies to TSER given the similarity of the presented results for both tasks for RIST, DrCIF and RDST.

6 Conclusions

We have shown that a transformation extracting random interval and shapelet subseries from different series representation can outperform individual interval and shapelet feature domain algorithms. While showing that hybrid algorithms have increased performance on the UCR archive is not new, RIST is much faster and simpler than other suggested hybrid approaches. The RIST transformation is fully unsupervised and can be easily applied to both classification and regression tasks. Our experiments show that the performance of RIST carries over to the TSER archive, presenting an early hybrid approach for TSER.

Given the performance of RIST for TSER, there is likely scope for further improvement by developing more sophisticated hybrid approaches for the task. HC2 outperforms RIST for classification, and as algorithms continue to be developed for the task a similar ensemble approach over multiple feature domains may find success as well.

While only briefly covered in our results, formatting the interval transform as a pipeline rather than an ensemble for the DrCIF algorithm resulted in significant scalability improvements. Even with the addition of the shapelet features, RIST is still much faster than the ensemble. This follows the approach of numerous recent algorithms, which produce a mass of features and leave a vector estimator to select the useful ones. While faster, a drawback of this approach is that it could be costly in terms of memory to perform these large transforms and require all features to be stored in memory at a single point.

Acknowledgements

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant number EP/W030756/1. The experiments were carried out on the High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia. We would like to thank all those responsible for helping maintain the time series dataset archives and those contributing to open source implementations of the algorithms.

Bibliography

[1] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[2] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[3] A. Benavoli, G. Corani, and F. Mangili. Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, 17:1–10, 2016.

[4] A. Bostrom and A. Bagnall. Binary shapelet transform for multiclass time series classification. *Transactions on Large-Scale Data and Knowledge Centered Systems*, 32:24–46, 2017.

[5] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[6] N. Cabello, E. Naghizade, J. Qi, and L. Kulik. Fast and accurate time series classification through supervised interval search. In *IEEE International Conference on Data Mining*, 2020.

[7] Nestor Cabello, Elham Naghizade, Jianzhong Qi, and Lars Kulik. Fast, accurate and interpretable time series classification through randomization. *arXiv preprint arXiv:2105.14876*, 2021.

[8] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77, 2018.

[9] H. Dau, A. Bagnall, K. Kamgar, M. Yeh, Y. Zhu, S. Gharghabi, C. Ratanamahatana, A. Chotirat, and E. Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[10] Angus Dempster, François Petitjean, and Geoffrey Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34:1454–1495, 2020.

[11] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. HYDRA: Competing convolutional kernels for fast and accurate time series classification. *arXiv preprint arXiv:2203.13652*, 2022.

[12] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[13] H. Deng, G. Runger, E. Tuv, and M. Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153, 2013.

[14] H. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. Webb, L. Idoumghar, P. Muller, and F. Petitjean. InceptionTime: finding

AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[15] M. Flynn, J. Large, and A. Bagnall. The contract random interval spectral ensemble (c-RISE): The effect of contracting a classifier on accuracy. In *proceedings of the Hybrid Artificial Intelligence Systems*, volume 11734 of *Lecture Notes in Computer Science*, pages 381–392. 2019.

[16] S. García and F. Herrera. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[17] Dominique Gay, Alexis Bondu, Vincent Lemaire, and Marc Boullé. Interpretable feature construction for time series extrinsic regression. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 804–816. Springer, 2021.

[18] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[19] David Guijo-Rubio, Matthew Middlehurst, Guilherme Arcencio, Diego Furtado Silva, and Anthony Bagnall. Unsupervised feature based algorithms for time series extrinsic regression. *arXiv preprint arXiv:2305.01429*, 2023.

[20] Antoine Guillaume, Christel Vrain, and Wael Elloumi. Random dilated shapelet transform: A new approach for time series shapelets. In *Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part I*, pages 653–664. Springer, 2022.

[21] Matthieu Herrmann, Chang Wei Tan, Mahsa Salehi, and Geoffrey I Webb. Proximity forest 2.0: A new effective and scalable similarity-based classifier for time series. *arXiv preprint arXiv:2304.05800*, 2023.

[22] J. Lines, L. Davis, J. Hills, and A. Bagnall. A shapelet transform for time series classification. In *proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[23] J. Lines, S. Taylor, and A. Bagnall. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions Knowledge Discovery from Data*, 12(5):1–36, 2018.

[24] C. Lubba, S. Sethi, P. Knaute, S. Schultz, B. Fulcher, and N. Jones. catch22: canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33(6):1821–1852, 2019.

[25] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, and G. Webb. Proximity forest: an effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery*, 33(3):607–635, 2019.

[26] M. Middlehurst, J. Large, G. Cawley, and A. Bagnall. The temporal dictionary ensemble (TDE) classifier for time series classification. In *proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 12457 of *Lecture Notes in Computer Science*, pages 660–676, 2020.

[27] Matthew Middlehurst and Anthony Bagnall. The freshprince: A simple transformation based pipeline time series classifier. In *International Con-*

ference on Pattern Recognition and Artificial Intelligence, pages 150–161. Springer, 2022.

[28] 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, 2020.

[29] 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*, 110:3211–3243, 2021.

[30] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *arXiv preprint arXiv:2304.13029*, 2023.

[31] Thach Le Nguyen and Georgiana Ifrim. Fast time series classification with random symbolic subsequences. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 50–65. Springer, 2022.

[32] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.

[33] P. Schäfer and U. Leser. Fast and accurate time series classification with WEASEL. In *proceedings of the ACM Conference on Information and Knowledge Management*, pages 637–646, 2017.

[34] Patrick Schäfer and Ulf Leser. Weasel 2.0 - a random dilated dictionary transform for fast, accurate and memory constrained time series classification. *arXiv preprint arXiv:2301.10194*, 2023.

[35] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I Webb. TS-CHIEF: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery*, 34(3):742–775, 2020.

[36] Chang Wei Tan, Christoph Bergmeir, Francois Petitjean, and Geoffrey Webb. Time series extrinsic regression. *Data Mining and Knowledge Discovery*, 35:1032—1060, 2021.

[37] Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey Webb. MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, 36:1623–1646, 2022.