TS-QUAD: A Smaller Elastic Ensemble For Time Series Classification With No Reduction In Accuracy

Jason Lines and George Oastler

University of East Anglia,
Norwich Research Park,
United Kingdom
{j.lines, g.oastler}@uea.ac.uk

Abstract. The Elastic Ensemble (EE) is a time series classification (TSC) ensemble that includes eleven nearest neighbour (NN) classifiers that use variations of eight elastic distance measures. While EE offers an accurate solution for TSC in the time domain, its relatively slow run-time is a weakness. This has led to new algorithms, such as Proximity Forest and TS-CHIEF, that have iterated on the design of EE by taking the same elastic measures and incorporating them into tree-based ensembles. These enhancements were implemented successfully and led to faster and more accurate time domain classifiers and, as such, development on the original EE algorithm subsided.
However, in this work we make the simple hypothesis that the original design of EE contains distance measures that capture the same discriminatory features, and as such, the ensemble includes redundant classifiers. If this were true, EE could perform to the same level in terms of accuracy with significantly less computation. If proven true this would have interesting implications to the design of algorithms such as Proximity Forest and TS-CHIEF that are based on the original EE implementation. To investigate this, we form a simple categorisation of the distance measures within EE and form four groups. We take one measure from each group, building an ensemble of four 1-NN classifiers that we call TS-QUAD: the Time Series QUARtet of distance-based classifiers. We demonstrate that this ensemble is able to match EE in terms of accuracy over 10 resamples of 85 datasets while containing fewer than 50% of the original EE constituents, implying that other elastic distance-based TSC ensembles could benefit from the design philosophy of TS-QUAD.

Keywords: time series · classification · elastic distance measures

1 Introduction

The Elastic Ensemble (EE) [12] is a time series classification (TSC) ensemble that combines eleven nearest neighbour (NN) classifiers built with eight distinct elastic distance measures. The motivation for creating EE was that, at the time,

the commonly used gold-standard for TSC was a 1-NN classifier coupled with Dynamic Time Warping (DTW) and a warping window parameter set through cross-validation. This led to variants of DTW being proposed in the literature, such as derivative [11] and weighted DTW [10], as well as other competing elastic distance measures such as Time Warp Edit (TWE) distance [15] and the Move-Split-Merge (MSM) distance [25]. While various approaches were proposed and evaluated in the literature, none significantly outperformed DTW 1-NN in terms of accuracy.

In [12], it was hypothesised that, even though these measures did not perform differently in terms of accuracy when combined with 1-NN classifiers, the measures themselves may detect similarity in different ways. It was proposed that combining classifiers built with each of these measures would detect a wider range of discriminatory features than using a single measure alone. EE was created to test this hypothesis by coupling the elastic measures each with a 1-NN classifier and combining predictions through a weighted voting scheme that was informed by training accuracy estimates. The results of experiments over the UCR datasets supported this hypothesis as EE outperformed all of its constituent classifiers, including DTW 1-NN, and all other TSC approaches that were published in the literature at the time.

Since EE was first proposed the field of TSC has grown rapidly and a range of diverse and effective algorithms have been introduced into the literature. Such algorithms include the Collective of Transformation-based Ensembles (COTE) [2], HIVE-COTE: the Hierarchical Vote Ensemble Collective of Transformation-based Ensembles [13], ROCKET [6], Proximity Forest (PF) [14], InceptionTime [7], and TS-CHIEF [24]. These algorithms are notable because each has now been shown to significantly outperform EE over the UCR datasets in terms of accuracy, but an interesting observation is that EE has been critical to the development of many successive state-of-the-art approaches. While ROCKET and InceptionTime are based on convolutional kernels and deep learning approaches respectively, PF, COTE and TS-CHIEF each incorporate the eight elastic distance measures that were first combined in EE and also use the same parameter options as proposed by EE, while the first version of HIVE-COTE contained EE itself as a constituent module to operate in the time domain.

It is clear that EE has influenced numerous algorithms, but development and refinement of the ensemble has all but ceased due to the relatively slow run-time of the nearest neighbour classifiers within EE. Efforts have been made to demonstrate that training and testing decisions can be significantly faster for EE through restricted neighbourhoods and randomised parameter searchers [19] but, in general, there was little need until now to revisit EE since subsequent algorithms such as PF are faster and more accurate than EE.

We do revisit the design of EE in this work however due to a simple observation that the original EE algorithm in [12] was designed to significantly outperform the gold standard at the time of DTW-1NN. While run-time was noted in this work, it was not a priority in the design of EE and it was not considered whether all distance measures were required in the final EE since its

introduction was a legitimate step-forward in the state of the art for TSC. Thus, we hypothesise that there may be underlying redundancy between some of the distance measures that were selected for EE and it is worth investigating. We believe that it is likely that we can build a subset of EE that will perform to the same level of accuracy as the whole ensemble while requiring far less computation. This finding would be of note because, while EE may not be widely used in TSC anymore, leading algorithms such as PF and TS-CHIEF are based on the original design of EE and may also include redundant distance measures. We choose to investigate this hypothesis with EE as it is a deterministic algorithm, unlike PF and TS-CHIEF that each include random selection, so it is clearer to demonstrate that the differences in performance are based on constituent selection alone rather than random chance.

We start by grouping the distance measures into four high-level and intuitive groupings and nominate a single measure from each group to include in a new subset of EE. We call this *TS-QUAD: the Time Series QUARtet of distance-based classifiers* for the purposes of this work, and results over 10 resamples of 85 UCR TSC problems demonstrate that TS-QUAD is no less accurate than EE while containing less than half of the constituent classifiers, and half of the original distance measures, of EE. This finding indicates that further research and refinement may be be possible of subsequent elastic TSC ensembles such as PF and TS-CHIEF.

2 Background and Related Work

We define a time series $T =< x_1, x_2, ..., x_m >$ as an ordered sequence of m real values. The ordering of attribute values is typically by units of time, but this is not a requirement. For example, electromagnetic spectroscopy readings are typically recorded in nanometres, not units of time, but we would consider this data a time series under our definition. Further, time series data can be univariate or multivariate depending on the number of dimensions or channels in the incoming data streams. In this work, we constrain our research efforts to the univariate case for classification of time series data.

For the supervised task of TSC, each series T_i must have an associated class label y_i. The objective of TSC is to use a set of n time series $\mathbf{T} = \{T_1, T_2, ..., T_n\}$, with associated class labels $Y = \{y_1, y_2, ..., y_n\}$, as training data to learn a function that maps from the space of all possible series to the space of all possible class labels. Then, when previously unseen series with unknown class labels are presented, predictions can be made to classify the unknown cases as one of the possible class values.

As alluded to earlier, TSC is a very active area of research and many diverse algorithms have been proposed and evaluated on a large number of TSC datasets (for example, [1] contains a large experimental comparison). We focus on EE as the starting point for this work, which performs classification in the time domain on raw series data, but it should be noted that a variety of other algorithms exist and work with discriminatory features that are discov-

ered in other transformation-based domains and the current state of the art for TSC in terms of accuracy, HIVE-COTE V2.0 [18] (HC2) is an ensemble that is formed with classifiers built over a range of different domains. Such individual domains include shapelet-based approaches that detect discriminatory features within phase-independent subsequences [26,9,8], interval-based approaches that focus on specific intervals within series [3,17], and histogram-based algorithms that extract features through counting the occurrence of repeated patterns to make classification decisions [22,23,16]. Recently, convolutional and deep learning approaches have shown promising results for TSC, such as InceptionTime [7] and ROCKET [6], while other approaches such as HIVE-COTE [13], HC2 and TS-CHIEF [24] are hybrid approaches that combine classifiers over multiple domains, such as the time, frequency and shapelet domains.

2.1 Classification in the Time Domain

Before HC2 and the other contemporary TSC algorithms were proposed, a large amount of research effort in the field focused on developing *elastic* distance measures to couple with nearest neighbour classifiers. The approach to perform classification in the time domain by measuring distances between series was popularised by the early success of using DTW with 1-NN classifiers and a warping window set through cross-validation (such as in [20,21]). Given the success of this approach, many subsequent efforts iterated on this design by proposing alternative time-series similarity measures to couple with 1-NN classifiers. These included variations of DTW, such as derivative [11] and weighted [10] DTW, and other specialised methods such as those that extended edit-distance approach to similarity (e.g. [4]) and hybrids based approaches such as TWE [15] and MSM [25]. While these proposed measure were often compared in experiments to DTW 1-NN, conclusions were anecdotal and no measure was demonstrated to significantly outperformed DTW-1NN over a large number of datasets.

2.2 The Elastic Ensemble (EE) and Extensions

EE was created to leverage from the wide range of elastic distance measures that had been introduced in the TSC literature in order to combine the different predictions of individual measures to produce a result that was more accurate than any approach in isolation. We will briefly reintroduce EE in this section, but to avoid retreading existing ground, we direct the interested reader to [12] for a more in-depth discussion and full implementation details of EE.

In total, EE contains eleven 1-NN classifiers that are coupled with versions of eight elastic distance measures. The first three 1-NN classifiers use measures that do not contain parameters to be set (Euclidean 1-NN, full window DTW 1-NN, full window derivative DTW 1-NN) and the remaining classifiers each use one of eight distance measures that require parameters to be set in training. The constituent classifiers in EE are summarised in Figure 1.

The elastic 1-NN classifiers in EE are combined through a weighted vote, where weights are established in training while parameters are optimised. Each

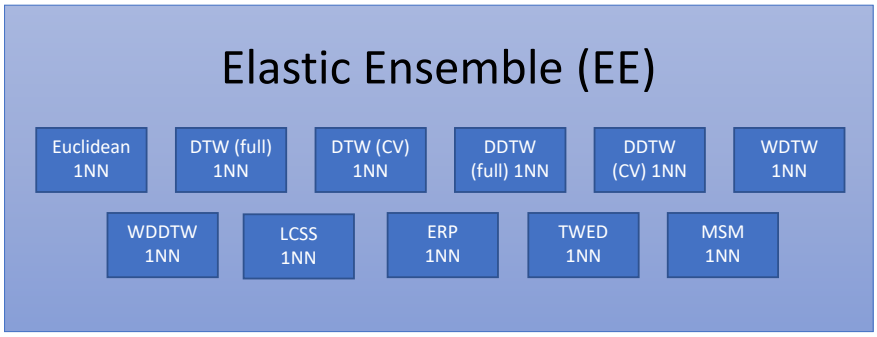

Fig. 1. A graphical representation of EE and the constituent classifiers that it contains. Eight of the eleven classifiers require distance measure parameters to be set in training, while the remaining three approaches (Euclidean, DTW full, DDTW full) do not require parameters to be set.

constituent is given 100 possible parameter options (which was originally motivated by DTW using windows in the range of 1%-100%) and leave-one-out cross-validation is used to determine which parameter setting performs best in training for each constituent classifier. The subsequent EE uses the best parameter options found in training for each constituent and also its corresponding training estimate to weight test predictions in a proportional vote. For example, if WDTW-1NN within the ensemble had a training accuracy of 87% for a given dataset, in testing, WDTW-1NN would be given a weight of 0.87 for its vote. By applying this weighting scheme to all constituent classifiers within EE, rather than using a simple majority vote, test classification over the UCR datasets [5] was significantly improved.

A clear downside of EE is that 1-NN classifiers are lazy classifiers, requiring an O(N) pass of the data for each classification decision. This is further slowed by the elastic measures each having O(m^2) run-time complexities for series of length m, and when combined with the training experiments that are required to find measure parameters and constituent voting weights, EE becomes a time-consuming algorithm to use. This has motivated work such as [14] where Proximity Forest (PF) was proposed as an improvement to EE. PF addressed these run-time issues by taking the eight distance measures from EE and making two key changes. First, 1-NN classifiers are not used, and they are replaced with tree-based classifiers that form an ensemble for test classification. Second, the same parameter ranges were considered for the elastic measures, but random selection is used when assessing potential splits within internal trees of PF. As a result, PF is much faster than EE in practice while still utilising the same elastic measures and parameter options. TS-CHIEF [24] is a further continuation of this research, using a similar structure and the same measures and parameter options as EE and PF in combination with trees that are built with features from other

transformation domains. Finally, it is worth noting that the underlying training scheme within EE has also been investigated, with [19] showing that over 90% of the time taken in training the standard EE algorithm could be skipped by using a random parameter grid-search and a reduced number of neighbours when comparing potential parameter options. However, a key observation that we leverage in this work is that none of these extensions consider whether *all* of the elastic distance measures within EE, PF and TS-CHIEF are required.

3 EE with fewer constituents: TS-QUAD

Our hypothesis for this investigation is that a number of internal classifiers within EE are replicating work, and through removing redundant learners, the resulting ensemble could make predictions with significantly less computation but no loss in accuracy. If this holds, it will have important implications for PF and TS-CHIEF; at its simplest, it would suggest that these classifiers could be built to the same level more quickly, as fewer parameter and measure combinations would need to be evaluated to produce an equivalent classifier. Importantly, however, there is also the possibility that this could lead to more accurate classifiers. The algorithm to build PF defaults to include 100 constituent tree classifiers; if the 100 internal learners are using complimentary measures and parameter options that detect the same discriminatory features, diversity within the ensemble will naturally be lower. However, with redundant measures reduced, the likelihood of the 100 internal learners being more diverse would be increased and this may lead to a more accurate ensemble overall, and this would also then translate to TS-CHIEF if true. However, investigating the effect on PF and TS-CHIEF is beyond the scope of this work as we wish to make a direct comparison between the inclusion and exclusion of constituent measures and classifiers. EE is a better choice for this goal as it is a deterministic algorithm and the differences between a full and reduced ensemble would not be explained by random chance. For this reason, we also do not include the clear speedups provided in [19] as this introduces randomness into the parameter selections for internal classifiers and the relative speedups for EE and a subset of EE would also be consistent.

To investigate whether we can remove constituent classifiers from EE we start by creating a simple, intuitive grouping of the elastic measures in Table 1. Our rationale for these groupings is that we do not believe that EE requires multiple measures that are designed to operate in similar manners. We wish to create an ensemble that only contains measures that have different design objectives, so we have created a high-level grouping that is based on the intuition behind each of the measures. We have also disregarded the full-window options for DTW and derivative DTW, as well as Euclidean distance, as these are already redundant if the parameterised versions of DTW and derivative DTW can recreate Euclidean distance and the full window equivalents in cases where those parameter options would be optimal.

The first group in Table 1, *time domain warping*, includes the classic DTW algorithm and weighted DTW (WDTW). The original DTW measure is applied

Table 1. The eight elastic distance measures first used together in EE placed into four high-level groupings. One measure was selected for TS-QUAD from each of the four groups and this is denoted in the table using ***.

Time domain warping	Derivative warping	Edit-distance	Hybrid measures
DTW	***DDTW****	***LCSS****	***MSM****
WDTW*	WDDTW	ERP	TWE

to raw time series, and WDTW is also applied to the raw data but uses weights to manipulate warping paths rather than a fixed cutoff. As DTW and WDTW are conceptually very similar and can result in identical distances with certain data and parameter options, they are clear candidates to group together.

Similarly, the second group in Table 1 contains derivative DTW (DDTW) and weighted derivative DDTW (WDDTW) into a *derivative warping* group. These measures are very similar to their origin measures, DTW and WDTW, and are both based on DTW with the variation that similarity is measured on the first-order derivatives of the time series, rather than directly on the raw data.

Thirdly, group three is titled *edit-distance* and includes the two distance measures from EE that are based around the idea of edit-distance. A full description of ERP and LCSS is given in [12], but briefly, these two measures are not based on DTW and instead measure the effort required to transform one series into another through operations such as additions, deletions and replacements. Edit-distance approaches are more common in other data mining applications, such as text mining, but have been successfully implemented for real-valued data by using threshold values and penalty functions.

Finally, the fourth group is named *hybrid measures* and it includes MSM and TWE. Both of these measures incorporate facets of time warping and edit-distance, and hence have been grouped together due to this high-level design similarity of incorporating characteristics from both the warping and edit-distance groups..

Our hypothesis is that including multiple measures from each of these groups in the same ensemble would introduce redundancy, rather than increasing diversity, and is therefore unnecessary computation. To test this, we select one distance measure from each group in Table 1. We do not wish to overfit the measure selections or introduce bias through looking at test results, so we use simple assumptions and practical knowledge, rather than classification accuracies, to make these decisions. From groups one and two we select WDTW and DDTW for use in our reduced ensemble. Our justification for using these two measures is slightly nuanced, but we believe it is more likely to result in diversity if one measure uses weighting and one uses a traditional warping window. We could select DTW and WDDTW, but we choose instead to select WDTW and DDTW as these two measures were the specific contributions of two TSC research papers [10,11], while WDDTW was a secondary contribution after the main WDTW measure and DTW was first used in other fields with WDTW posed as an improvement upon it. We do not expect this rationale to make a

large difference overall however, and it is likely that using DTW and WDDTW would result in similar coverage to using WDTW and DDTW if our hypothesis is correct. For groups three and four, our decisions are simpler; timing experiments were carried out in [12] and demonstrated that LCSS was faster than ERP, and MSM was faster than TWE on the same data. We use this prior knowledge to select LCSS and MSM respectively for convenience as these timing results can be recreated simply without introducing bias or observing any results from real data. Since our hypothesis is that the discriminatory features captured by measures within the same group will be consistent regardless of run-time it is therefore sensible to prioritise faster measures.

Our final ensemble contains four elastic distance measures: WDTW, DDTW, LCSS and MSM. Each is combined with a 1-NN classifier and form part of a smaller elastic ensemble. For the purposes of this work, we call this new ensemble TS-QUAD (**T**ime-**S**eries **QUA**rtet of **D**istance-based classifiers). We do not expect TS-QUAD to compete with the state of the art, but in this work it will help to either support or refute our hypothesis that EE, PF and TS-CHIEF contain redundant distance measures that could be removed without reducing accuracy.

4 Experimental Procedure

We compare TS-QUAD to EE over 10 resamples of the UCR TSC problems [5] using the same 85-dataset version of the repository that has been widely used in recent work [1,13,14,24]. As discussed previously, the primary motivation for this research is not to outperform state-of-the-art algorithms such as HIVE-COTE with TS-QUAD, but rather the motivation is to demonstrate that four elastic measures can perform as well together as the full set of eight that are used by EE. To this end we compare TS-QUAD directly to EE in our experiments. The datasets are resampled using the same random seeds as the first 10 resamples in [1] to ensure that results are reproducible and comparable (with the first 'resample' being the default train/test split of the data), and we also use the same implementations of the distance measures and 1-NN classifiers that were originally used for EE in [12] to ensure that there are no differences caused by inconsistent implementations. The source code for the distance measures and classifiers is freely available in the open source Java toolkit `tsml` and can be found here[1], while the code to create consistent resamples of the dataset can also be found within the same toolkit here[2].

[1] `https://github.com/uea-machine-learning/tsml/tree/master/src/main/java/tsml/classifiers/legacy/elastic_ensemble`

[2] `https://github.com/uea-machine-learning/tsml/blob/master/src/main/java/utilities/InstanceTools.java`

5 Results

The results of EE and TS-QUAD over 10 resamples of the 85 UCR datasets are summarised in Table 2 and the full results are given in Table 3. It can be seen from the summarised results that there is very little to choose from between TS-QUAD and EE in terms of both average accuracy and average rank. EE has a slightly superior rank over the 85 problems, with 1.494 versus 1.506, while TS-QUAD in fact has a higher average accuracy than EE with 81.16% and 80.89% over these experiments.

Table 2. The average accuracies and ranks of EE and TS-QUAD over the 85 UCR datasets. The accuracies are averaged over 10 resamples, and the average rank is calculated by first ranking each classifier on their respective average accuracy for a given dataset, and then averaging the ranks across all 85 datasets. Overall, EE won on 42 datasets, TS-QUAD on 41, and they tied on two.

	EE	**TS-QUAD**
Average Accuracy	80.89%	**81.16%**
Average Rank	**1.494**	1.506

There is no significant difference in accuracy between EE and TS-QUAD, confirmed by both a paired t-test and a Wilcoxon signed-rank test. This result is a very positive indication that our original hypothesis holds true and that we do not need to use all eight distance measures that were originally combined in [12] to produce a competitive ensemble of elastic-based 1-NN classifiers. While the results of TS-QUAD do not challenge the state of the art, and were never expected to, they do suggest that further investigation is required to verify whether this finding is true when based to other time series ensembles that built upon the design principles established by the introduction of EE, most notably PF and TS-CHIEF. We also note that it may be possible to further improve upon TS-QUAD by optimising the constituent measures that are included and it would likely be possible to post-process all combinations of the EE constituents to produce a more accurate subset. However, it would not be constructive to optimise the ensemble in this way as it would likely lead to overfitting on the UCR datasets specifically. We believe TS-QUAD is a fair subset of EE, as it is based on an intuitive and high-level grouping of measures, but it is not intended to be solution to TSC problems itself. Its main purpose is to motivate further research effort in the area of elastic measure selection and we believe these results achieve this goal.

6 Conclusions, future work and extensions

In this work we have investigated whether the Elastic Ensemble (EE) [12] contains distance measures that capture overlapping discriminatory features. We

Table 3. Average accuracies over the 85 UCR TSC problems for EE and TS-QUAD. The accuracies reported are averaged over 10 resamples of each dataset (please note that some of the dataset names have been shortened for presentation, but each dataset is identical to those used in other work such as [1].

Dataset	EE	TS-QUAD	Dataset	EE	TS-QUAD
Adiac	**67.16**	66.45	MedicalImages	76	**76.38**
ArrowHead	86.06	**86.17**	MiddlePhalanxAge	59.55	**64.55**
Beef	56	**58.67**	MiddlePhalanxCorrect	78.11	**78.63**
BeetleFly	77.5	**78**	MiddlePhalanxTW	51.56	**54.68**
BirdChicken	**86.5**	83	MoteStrain	87.26	**87.83**
CBF	98.59	**99.39**	NonInvasiveFT	**84.94**	83.4
Car	80.83	**82.83**	NonInvasiveFT2	**91.39**	90.35
ChlorineConcentration	66.43	**68.93**	OSULeaf	**81.9**	80.95
CinCECGTorso	94.62	**95.78**	OliveOil	87	**87.67**
Coffee	**98.21**	97.86	PhalangesCorrect	77.82	**78.64**
Computers	72.32	**73.64**	Phoneme	**30.16**	28.41
CricketX	**81.08**	79.54	Plane	**100**	99.81
CricketY	**78.9**	77.26	ProximalPhalanxAgeGroup	79.71	**82.93**
CricketZ	**80.31**	79.38	ProximalPhalanxCorrect	82.99	**83.81**
DiatomSizeRed.	94.87	**95.75**	ProximalPhalanxTW	75.95	**77.56**
DistalPhalanxAge	74.68	**75.25**	RefrigerationDevices	65.33	**67.95**
DistalPhalanxCorrect	**76.23**	75.87	ScreenType	55.73	**56.24**
DistalPhalanxTW	65.25	**66.83**	ShapeletSim	82.72	**91.22**
ECG200	**89.2**	88.6	ShapesAll	88.5	**87.97**
ECG5000	93.68	**93.8**	SmallKitchenAppliances	69.55	**70.24**
ECGFiveDays	85.39	**88.58**	SonyAIBORobotSurface1	78.49	**79.63**
Earthquakes	73.17	**73.53**	SonyAIBORobotSurface2	88.61	**88.91**
ElectricDevices	**81.43**	81.41	StarlightCurves	93.92	**94.62**
FaceAll	96.6	**97.07**	Strawberry	95.57	**95.59**
FaceFour	86.48	**90.91**	SwedishLeaf	**91.98**	91.46
FacesUCR	94.83	**96.68**	Symbols	**95.58**	94.32
FiftyWords	**82.31**	81.85	SyntheticControl	**99.4**	99.07
Fish	**91.49**	90.97	ToeSegmentation1	**77.68**	77.5
FordA	73.74	**74.9**	ToeSegmentation2	**90**	89.92
FordB	**74.95**	74.94	Trace	**99.5**	**99.5**
GunPoint	**96.87**	96.8	TwoLeadECG	**95.49**	94.14
Ham	73.52	**74.1**	TwoPatterns	**100**	99.99
HandOutlines	**88.62**	88.22	UWaveGestureLibraryAll	**96.94**	96.71
Haptics	**44.19**	43.93	UWaveGestureLibraryX	**80.63**	79.04
Herring	57.03	**58.13**	UWaveGestureLibraryY	**72.95**	71.21
InlineSkate	**47.44**	46.64	UWaveGestureLibraryZ	**72.49**	70.97
InsectWingbeatSound	**57.66**	57.48	Wafer	99.71	**99.73**
ItalyPowerDemand	**95.27**	95.07	Wine	**85.74**	**85.74**
LargeKitchenAppliances	**81.68**	80.69	WordSynonyms	**77.63**	77.26
Lightning2	82.95	**83.11**	Worms	**63.38**	62.47
Lightning7	**75.07**	71.78	WormsTwoClass	**72.08**	70.91
Mallat	**95.95**	95.45	Yoga	88.47	**88.5**
Meat	**97.83**	97.33			

hypothesised that a number of measures within EE were redundant, and that we could therefore form a subset of constituents from EE that would perform no worse under experimental conditions in terms of accuracy but with significantly less computation. We formed this subset by first grouping each of the distance measures from EE together into simple and intuitive categories. We proposed four categories and used one elastic measure from each group, in combination with a 1-NN classifier, to form TS-QUAD. TS-QUAD contains four internal 1-NN classifiers, rather than the eleven (built with eight distance measures) that are contained within EE. We demonstrated that, over 10 resamples of 85 datasets, there is no significant difference in terms of accuracy when comparing EE to TS-QUAD. This work has demonstrated that it is indeed possible to perform as well as the full EE while only using half of the original distance measures that were contained by the full ensemble. This finding suggests that future work should be conducted to investigate whether similar improvements could be made to algorithms that are informed by the original design of EE, such as PF and TS-CHIEF, and this may lead to faster and more accurate elastic-based TSC ensembles.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: 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. Bagnall, A., Lines, J., Hills, J., Bostrom, A.: Time-series classification with COTE: The collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering **27**, 2522–2535 (2015)
3. Baydogan, M., Runger, G., Tuv, E.: A bag-of-features framework to classify time series. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(11), 2796–2802 (2013)
4. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: proceedings of 30th International Conference on Very Large Databases (VLDB) (2004)
5. Dau, H., Bagnall, A., Kamgar, K., Yeh, M., Zhu, Y., Gharghabi, S., Ratanamahatana, C., Chotirat, A., Keogh, E.: The UCR time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6), 1293–1305 (2019)
6. Dempster, A., Petitjean, F., Webb, G.: ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery **34**, 1454–1495 (2020)
7. Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D., Weber, J., Webb, G., Idoumghar, L., Muller, P., Petitjean, F.: InceptionTime: finding AlexNet for time series classification. Data Mining and Knowledge Discovery **34**(6), 1936–1962 (2020)
8. Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning time-series shapelets. In: proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014)
9. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery **28**(4), 851–881 (2014)

10. Jeong, Y., Jeong, M., Omitaomu, O.: Weighted dynamic time warping for time series classification. Pattern Recognition **44**, 2231–2240 (2011)
11. Keogh, E., Pazzani, M.: Derivative dynamic time warping. In: proceedings of 1st SIAM International Conference on Data Mining (2001)
12. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery **29**, 565–592 (2015)
13. Lines, J., Taylor, S., Bagnall, A.: 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)
14. Lucas, B., Shifaz, A., Pelletier, C., O'Neill, L., Zaidi, N., Goethals, B., Petitjean, F., Webb, G.: Proximity forest: an effective and scalable distance-based classifier for time series. Data Mining and Knowledge Discovery **33**(3), 607–635 (2019)
15. Marteau, P.: Time warp edit distance with stiffness adjustment for time series matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 306–318 (2009)
16. Middlehurst, M., Large, J., Cawley, G., Bagnall, A.: 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. Lecture Notes in Computer Science, vol. 12457, pp. 660–676 (2020)
17. Middlehurst, M., Large, J., Bagnall, A.: The canonical interval forest (CIF) classifier for time series classification. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 188–195. IEEE (2020)
18. Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., Bagnall, A.: Hive-cote 2.0: a new meta ensemble for time series classification. Machine Learning **Online First**, 1–33 (2021), `http://link.springer.com/article/10.1007/s10994-021-06057-9`
19. Oastler, G., Lines, J.: A significantly faster elastic-ensemble for time-series classification. In: proceedings of Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, vol. 11871, pp. 446–453 (2019)
20. Rakthanmanon, T., Bilson, J., Campana, L., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. ACM Transactions on Knowledge Discovery from Data **7**(3) (2013)
21. Rath, T., Manamatha, R.: Word image matching using dynamic time warping. In: proceedings of Computer Vision and Pattern Recognition (2003)
22. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery **29**(6), 1505–1530 (2015)
23. Schäfer, P., Leser, U.: Fast and accurate time series classification with WEASEL. In: proceedings of the ACM on Conference on Information and Knowledge Management. pp. 637–646 (2017)
24. Shifaz, A., Pelletier, C., Petitjean, F., Webb, G.I.: TS-CHIEF: a scalable and accurate forest algorithm for time series classification. Data Mining and Knowledge Discovery **34**(3), 742–775 (2020)
25. Stefan, A., Athitsos, V., Das, G.: The Move-Split-Merge metric for time series. IEEE Transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2013)
26. Ye, L., Keogh, E.: Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery **22**(1-2), 149–182 (2011)