Data Min Knowl Disc
DOI 10.1007/s10618-014-0361-2

Time series classification with ensembles of elastic distance measures

Jason Lines · **Anthony Bagnall**

Received: 10 January 2014 / Accepted: 5 June 2014

Abstract Several alternative distance measures for comparing time series have recently been proposed and evaluated on time series classification (TSC) problems. These include variants of dynamic time warping (DTW), such as weighted and derivative DTW, and edit distance-based measures, including longest common subsequence, edit distance with real penalty, time warp with edit, and move–split–merge. These measures have the common characteristic that they operate in the time domain and compensate for potential localised misalignment through some elastic adjustment. Our aim is to experimentally test two hypotheses related to these distance measures. Firstly, we test whether there is any significant difference in accuracy for TSC problems between nearest neighbour classifiers using these distance measures. Secondly, we test whether combining these elastic distance measures through simple ensemble schemes gives significantly better accuracy. We test these hypotheses by carrying out one of the largest experimental studies ever conducted into time series classification. Our first key finding is that there is no significant difference between the elastic distance measures in terms of classification accuracy on our data sets. Our second finding, and the major contribution of this work, is to define an ensemble classifier that significantly outperforms the individual classifiers. We also demonstrate that the ensemble is more accurate than approaches not based in the time domain. Nearly all TSC papers in the data mining literature cite DTW (with warping window set through cross validation) as the benchmark for comparison. We believe that our ensemble is the first ever classifier to significantly outperform DTW and as such raises the bar for future work in this area.

Responsible editor: Eamonn Keogh.

J. Lines (✉) · A. Bagnall
University of East Anglia, Norwich, UK
e-mail: j.lines@uea.ac.uk

A. Bagnall
e-mail: anthony.bagnall@uea.ac.uk

Keywords Time series classification · Elastic distance measures · Ensembles

1 Introduction

Time-series classification (TSC) problems involve training a classifier on a set of cases, where each case contains an ordered set of real valued attributes and a class label. Time-series classification problems arise in a wide range of fields including, but not limited to, data mining, statistics, machine learning, signal processing, environmental sciences, computational biology, image processing and chemometrics. A wide range of algorithms have been proposed for solving TSC problems [see, for example (Ding et al. 2008; Rakthanmanon and Keogh 2013; Batista et al. 2011; Jeong et al. 2011; Bagnall et al. 2012; Deng et al. 2013; Hu et al. 2013)].

In Bagnall et al. (2012), Bagnall et al. argue that the easiest way to gain improvement in accuracy on TSC problems is to transform into an alternative data space where the discriminatory features are more easily detected. They constructed classifiers on data in the time, frequency, autocorrelation and principal component domains and combined predictions through alternative ensemble schemes. The main conclusion in Bagnall et al. (2012) is that for problems where the discriminatory features are based on similarity in change and similarity in shape, operating in a different data space produces a better performance improvement than designing a more complex classifier for the time domain. However, the issue of what is the best technique in a single data domain is not addressed in Bagnall et al. (2012). Our aim is to experimentally determine the best method for constructing classifiers in the time domain, an area that has drawn most of the attention of TSC data mining researchers (Batista et al. 2011; Jeong et al. 2011). The general consensus amongst data mining researchers is that *"simple nearest neighbor classification is very difficult to beat"* (Batista et al. 2011), and the majority of research has concentrated on alternative distance measures that can compensate for localised distortion, such as dynamic time warping (DTW) and longest common subsequence (LCSS) distances. In light of extensive experimental evidence (Ding et al. 2008), DTW with warping window size set through cross validation is commonly accepted as the gold standard against which to compare alternative measures.

Recently, a number of new elastic measures have been proposed that are variations of the time warp and edit distance approaches. A version of DTW that weights against large warpings (WDTW) is described in Jeong et al. (2011). The weighting scheme can be used in conjunction with dynamic time warping and an alternative version based on first order differences (DDTW), described in Keogh and Pazzani (2011). Variants on the edit distance approach have also been introduced, including edit distance with real penalty (ERP) (Chen et al. 2005), time warp edit (TWE) distance (Marteau 2009), and the move–split-merge (MSM) distance metric (Stefan et al. 2012).

Our first objective is to extend the work in Ding et al. (2008) to statistically test whether these two new measures are in fact significantly better than DTW. Through statistical rank based tests (Demšar 2006) on results from 75 data sets we find that there is no significant difference in accuracy between the elastic measures (when they are used as a distance function for a nearest neighbour classifier).

This leads us to the question, should we just use DTW and ignore the more complex, less intuitive measures? Through a comparison of the predictions on the training data, we find that although there is no overall difference in accuracy, there is frequently significant difference between the classifiers when it comes to actual predictions. This leads us to believe that by combining the predictions of these classifiers we may be able to achieve an overall improvement in accuracy. We propose four alternative ensemble schemes for combining predictions from the different classifiers. Our primary contribution is to demonstrate that the accuracy of the ensembles is significantly better than the accuracy of any single classifier. We demonstrate that the elastic ensemble results are better than any other comparable results we have found in the literature, including classifiers built with shapelets, intervals or pattern frequency histograms. The ensemble is, we believe, the first algorithm to significantly outperform DTW (with window size set through cross validation) on the UCR data.

We have conducted over three million experiments on 75 TSC problems. 46 datasets come from the UCR repository (Keogh et al. 2011), 24 problems are from other published research, including (Bagnall 2012; Rakthanmanon and Keogh 2013), and 5 are new data sets on electricity device classification problems derived from a UK government sponsored study (Trust 2012). All datasets and code to reproduce experiments and results are available online (Lines and Bagnall 2014).

The structure of this paper is as follows. In Sect. 2 we provide background into TSC and review related research on elastic distance measures and classifiers using alternative data representations. In Sect. 3 we detail the 75 data sets we used in experiments. In Sect. 4 we describe the ensemble schemes we use to combine the alternative elastic distance measures. In Sect. 5 we present an empirical comparison of elastic distance measures, and in Sect. 6 we present and analyse the ensemble results and compare them with the current state-of-the-art in TSC. Finally, in Sect. 7 we summarise our conclusions and discuss future work.

2 Background and related work

2.1 Time Series Classification

We define time series classification as the problem of building a classifier from a collection of labelled training time series. We limit our attention to problems where each time series has the same number of observations. We define a time series $\mathbf{x_i}$ as a set of ordered observations of length m and an associated class label y_i. The training set is a set of n labelled pairs $(\mathbf{x_i}, y_i)$. For traditional classification problems, the order of the attributes is unimportant and the interaction between variables is considered independent of their relative positions. For time series data, the ordering of the variables is often crucial in finding the best discriminating features.

The majority of the data mining research into TSC has concentrated on similarity in time. This can be quantified by measures such as Euclidean distance or correlation. Similarity in time is characterised by the situation where the series from each class are observations of an underlying common curve in the time dimension. Variation around this underlying common shape is caused by noise in observation, and also by possible misalignment in the time axes that may cause a slight phase shift. Elastic measures that allow for some realignment can compensate for this shifting.

2.2 Dynamic Time Warping

For similarity in shape, DTW is commonly used to mitigate against distortions in the time axis (Ratanamahatana and Keogh 2005). Suppose we want to measure the distance between two series, $\mathbf{a} = \{a_1, a_2, \ldots, a_m\}$ and $\mathbf{b} = \{b_1, b_2, \ldots, b_m\}$. Let $M(\mathbf{a}, \mathbf{b})$ be the $m \times m$ pointwise distance matrix between $\mathbf{a}$ and $\mathbf{b}$, where $M_{i,j} = (a_i - b_j)^2$.

A warping path

$$P =< (e_1, f_1), (e_2, f_2), \ldots, (e_s, f_s) >$$

is a set of points (i.e. pairs of indexes) that define a traversal of matrix M. So, for example, the Euclidean distance $d_E(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m}(a_i - b_i)^2$ is the path along the diagonal of M.

A valid warping path must satisfy the conditions $(e_1, f_1) = (1, 1)$ and $(e_s, f_s) = (m, m)$ and that $0 \leq e_{i+1} - e_i \leq 1$ and $0 \leq f_{i+1} - f_i \leq 1$ for all $i < m$.

The DTW distance between series is the path through M that minimizes the total distance, subject to constraints on the amount of warping allowed. Let $p_i = M_{a_{e_i}, b_{f_i}}$ be the distance between elements at position e_i of $\mathbf{a}$ and at position f_i of $\mathbf{b}$ for the i^{th} pair of points in a proposed warping path P. The distance for any path P is

$$D_P(\mathbf{a}, \mathbf{b}) = \sum \_i = 1^s p_i.$$

If $\mathcal{P}$ is the space of all possible paths, the DTW path P^* is the path that has the minimum distance, i.e.

$$P^* = \min_{P \in \mathcal{P}}(D_P(\mathbf{a}, \mathbf{b})),$$

and hence the DTW distance between series is

$$D_{P*}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{k} p_i.$$

The optimal path P^* can be found exactly through a dynamic programming formulation. This can be a time consuming operation, and it is common to put a restriction on the amount of warping allowed. This restriction is equivalent to putting a maximum allowable distance between any pairs of indexes in a proposed path. If the warping window, r, is the proportion of warping allowed, then the optimal path is constrained so that

$$|e_i - f_i| \leq r \cdot m \quad \forall (e_i, f_i) \in P^*.$$

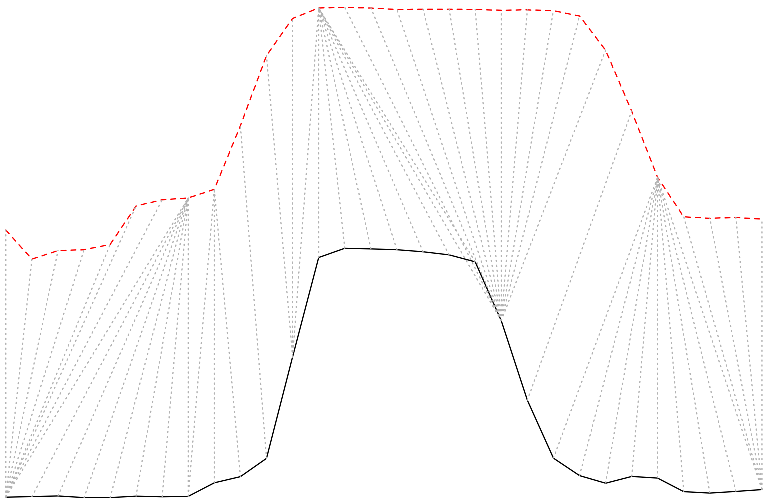

Fig. 1 An example of unconstrained DTW with two series from the GunPoint dataset (Keogh et al. 2011). DTW mitigates the misalignment of the series by matching the peaks and troughs of the data

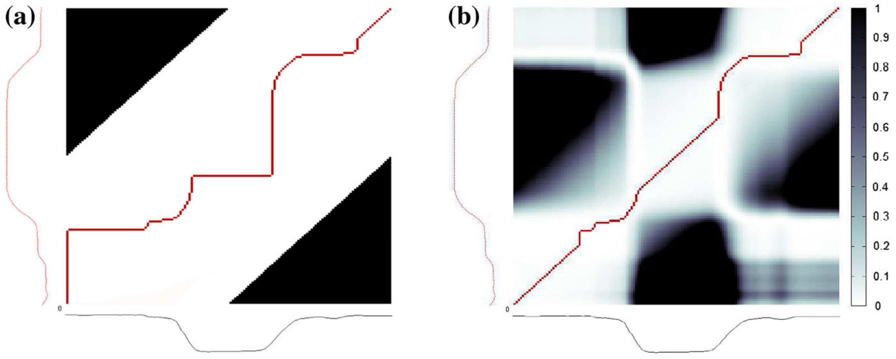

Fig. 2 Example distance matrices for the two GunPoint series from Fig. 1 with warping paths for a 50 % warping window with DTW (**a**) and WDTW with a penalty value of 0.1 (**b**). The *dark areas* in **a** depict the limits of the warping window where the path may not pass, whereas in **b** the *dark areas* represent areas that are highly weighted. It can be seen that the gradient of the weighting fuction encourages the path to avoid highly weighted areas in **b**, but it does not strictly prevent traversal of those areas as a warping window does

2.3 Derivative Dynamic Time Warping

Keogh and Pazzani proposed a modification of DTW called DDTW (Keogh and Pazzani 2011) that first transforms the series into a series of first order differences. The motivation for DDTW was to introduce a measure that avoids *singularities*, where a single point on one series may map onto a large subsection of another time series and create pathological results. Given a series $\mathbf{a} = \{a_1, a_2, \ldots, a_m\}$, the difference series is $\mathbf{a}' = \{a'_1, a'_2, \ldots, a'_{m-1}\}$ where a'_i is defined as the average of the slopes between a_{i-1} and a_i and a_i and a_{i+1}, i.e.

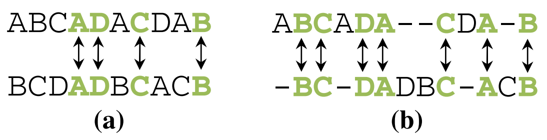

Fig. 3 An example of the LCSS problem. The example in **a** shows a pairwise matching of the two strings, while **b** demonstrates an alignment that allows shifting within the strings to allow for more matches to be made. The example in **a** is analogous with the LCSS distance with no elasticity, whereas **b** represents the LCSS distance with full elasticity. This can be controlled using a parameter akin to the warping window in DTW

$$a'_i = \frac{(a_i - a_{i-1}) + (a_{i+1} - a_{i-1})/2}{2},$$

for $1 < i < m$. DDTW is designed to mitigate against noise in the series that can adversely affect DTW, and has also be used in conjunction with standard DTW to simultaneously calculate similarity between series (Górecki and Łuczak 2013).

2.4 Weighted Dynamic Time Warping

A weighted form of DTW (WDTW) was proposed by Jeong et al. (2011). WDTW adds a multiplicative weight penalty based on the warping distance between points in the warping path. It favours reduced warping, and is a smooth alternative to the cutoff point approach of using a warping window. When creating the distance matrix M, a weight penalty $w_{|i-j|}$ for a warping distance of $|i - j|$ is applied, so that

$$M_{i,j} = w_{|i-j|}(a_i - b_j)^2.$$

A logistic weight function is proposed in Jeong et al. (2011), so that a warping of a places imposes a weighting of

$$w(a) = \frac{w_{max}}{1 + e^{-g\cdot(a-m/2)}},$$

where $w_{\max}$ is an upper bound on the weight (set to 1), m is the series length and g is a parameter that controls the penalty level for large warpings. The larger g is, the greater the penalty for warping (Fig. 2).

2.5 Longest Common Subsequence

The LCSS distance is based on the solution to the LCSS problem in pattern matching. The typical problem is to find the longest subsequence that is common to two discrete series based on the edit distance. An example using strings is shown in Fig. 3.

This approach can be extended to consider real-valued time series by using a distance threshold ϵ, which defines the maximum difference between a pair of values that is allowed for them to be considered a match. LCSS finds the optimal alignment between two series by inserting gaps to find the greatest number of matching pairs.

The LCSS between two series **a** and **b** can be found using Algorithm 1.

Algorithm 1 LCSS ($\mathbf{a}, \mathbf{b}$)

```
1: Let $L$ be an $(m+1) \times (m+1)$ matrix initialised to zero.
2: for $i \leftarrow m$ to 1 do
3:   for $j \leftarrow m$ to 1 do
4:     $L_{i,j} \leftarrow L_{i+1,j+1}$
5:     if $a_i = b_j$ then
6:       $L_{i,j} \leftarrow L_{i,j} + 1$
7:     else if $L_{i,j+1} > L_{i,j}$ then
8:       $L_{i,j} \leftarrow L_{i,j+1}$
9:     else if $L_{i+1,j} > L_{i,j}$ then
10:      $L_{i,j} \leftarrow L_{i+1,j}$
11: return $L_{1,1}$
```

The LCSS distance between **a** and **b** is

$$d_{\text{LCSS}}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\text{LCSS}(\mathbf{a}, \mathbf{b})}{m}.$$

2.6 Edit Distance with Real Penalty

Other edit distance-based similarity measures have also been proposed. One such approach is edit distance on real sequences (EDR) (Chen and Ng 2004). Like LCSS, EDR uses a distance threshold to define when two elements of a series match, but also includes a constant penalty that is applied for non-matching elements and where gaps are inserted to create optimal alignments. However EDR does not satisfy triangular inequality, as equality is relaxed by assuming elements are equal when the distance between them is less than or equal to ϵ. This was revised in Chen et al. (2005), where ERP was introduced. The motivation for ERP is that it is a metric as it satisfies triangular inequality by using 'real penalty', which uses the distance between elements when there is no gap and a constant value g for when gaps occur. The ERP distance between element i of series a and element j of series b is

$$ERP(a_i, b_j) = \begin{cases} |a_i - b_j| & \text{if } |a_i - b_j| \leq \epsilon \\ |a_i - g| & \text{if } b_j \text{ is a gap} \\ |b_j - g| & \text{if } a_i \text{ is a gap,} \end{cases}$$

and the full ERP distance between series a of length m and series b of length n is given recursively as

$$d_{ERP}(a, b) = \begin{cases} \sum_{i=1}^{n} |b_i - g| & \text{if } \; m = 0 \\ \sum_{i=1}^{m} |a_i - g| & \text{if } \; n = 0 \\ min\{d_{ERP}(Tail(a), Tail(b)\} + ERP(a_1, b_1), & \text{otherwise} \\ \qquad d_{ERP}(Tail(a), b) + ERP(a_1, gap), & \\ \qquad d_{ERP}(a, Tail(b)) + ERP(gap, b_1)\}, & \end{cases}$$

where $Tail(a) = \{a_2, a_3, ..., a_m\}$.

2.7 Time warp edit distance (TWED)

Introduced by Marteau (2009), TWE distance is an elastic distance measure that, unlike DTW and LCSS, is also a metric. It encompasses characteristics from both LCSS and DTW as it allows warping in the time axis and combines the edit distance with *Lp-norms*. The warping, called *stiffness*, is controlled by a parameter ν. Unlike a warping window that constrains a DTW search, stiffness enforces a multiplicative penalty on the distance between matched points. Setting $\nu = 0$ results in no stiffness, or *null stiffness*, giving a distance measure equivalent to a full DTW search. Setting $\nu = \infty$ gives Euclidean distance. TWED redefines the *insert*, *remove* and *match* operations used in edit distance, in favour of *delete_a*, *delete_b* and *match*. The *delete_a* operation occurs when an element is removed from the first series to match the second, and *delete_b* is when an element of the second series is removed to match the first. An *Lp-norm* distance calculation is used when matches are found, and a constant penalty value λ is applied when sequences do not match. The formal definition TWED can be found in Marteau (2009), and a dynamic programming implementation is given in Algorithm 2.

Algorithm 2 TWE Distance($\mathbf{a}$, $\mathbf{b}$, λ, ν)

```
1: Let $D$ be an $m + 1 \times n + 1$ matrix initialised to zero.
2: $D(1, 1) = 0$
3: $D(2, 1) = {a_1}^2$
4: $D(1, 2) = {b_1}^2$
5: for $i \leftarrow 3$ to $m + 2$ do
6:     $D(i, 1) = D(i - 1, 1) + (a_{i-2} - a_{i-1})^2$
7: for $j \leftarrow 3$ to $n + 2$ do
8:     $D(1, i) = D(1, j - 1) + (b_{j-2} - b_{j-1})^2$
9: for $i \leftarrow 2$ to $m + 2$ do
10:    for $j \leftarrow 2$ to $n + 2$ do
11:        if $i > 1$ and $j > 1$ then
12:            $dist1 = D(i - 1, j - 1) + \nu \times |i - j| \times 2 + (a_{i-1} - b_{j-1})^2 + (a_{i-2} - b_{j-2})^2$
13:        else
14:            $dist1 = D(i - 1, j - 1) + \nu \times |i - j| + (a_{i-1} - b_{j-1})^2$
15:        if $i > 1$ then
16:            $dist2 = D(i - 1, j) + (a_{i-1} - a_{i-2})^2 + \lambda + \nu$
17:        else
18:            $dist2 = D(i - 1, j) + {a_{i-1}}^2 + \lambda$
19:        if $j > 1$ then
20:            $dist3 = D(i, j - 1) + (b_{j-1} - b_{j-2})^2 + \lambda + \nu$
21:        else
22:            $dist3 = D(i, j - 1) + {b_{j-1}}^2 + \lambda$
23:        $D(i, j) = \min(dist1, dist2, dist3)$
24: return $D(m + 1, n + 1)$
```

2.8 Move-Split-Merge

MSM was introduced in Stefan et al. (2012). The authors motivate the introduction of MSM as it satisfies a number of desirable traits that they set out to incorporate into a single similarity measure: it is robust to temporal misalignments; it is translation invariant; it has a competitive quadratic run-time with DTW; and it is a metric. MSM is conceptually similar to other edit distance-based approaches, where similarity is calculated by using a set of operations to transform a given series into a target series. Each operation has an associated cost, and three operations are defined for MSM: move, split, and merge. Move is synonymous with a substitute operation, where one value is replaced by another. Split and merge differ from other approaches, as they attempt to add context to insertions and deletions. Stefan et al. state that cost of inserting and deleting values should depend on the value itself and adjacent values, rather than treating all insertions and deletions equally (for example, as in ERP). Therefore, the split operation is introduced to insert an identical copy of a value immediately after itself, and the merge operation is used to delete a value if it directly follows an identical value. The formal definition of MSM can be found in Stefan et al. (2012), and a dynamic programming implementation is given in Algorithm 3 using the cost function

$$C(a_i, a_{i-1}, b_j) = \begin{cases} c \textbf{ if } a_{i-1} \leq a_i \leq b_j \textbf{ or } a_{i-1} \geq a_i \geq b_j \\ c + \min(|a_i - a_{i-1}|, |a_i - b_j|) \textbf{ otherwise}. \end{cases}$$

Algorithm 3 MSM Distance(**a**, **b**)

```
1: Let $D$ be an $m \times n$ matrix initialised to zero.
2: $D(1, 1) = |a_1 - b_1|$
3: for $i \leftarrow 2$ to $m$ do
4:     $D(i, 1) = D(i - 1, 1) + C(a_i, a_{i-1}, b_1)$
5: for $i \leftarrow 2$ to $n$ do
6:     $D(1, i) = D(1, i - 1) + C(b_i, a_1, b + i - 1)$
7: for $i \leftarrow 2$ to $m$ do
8:     for $j \leftarrow 2$ to $n$ do
9:         $D(i, j) = \min(D(i - 1, j - 1) + |a_i - b_j|,$
                       $D(i - 1, j) + C(a_i, a_{i-1}, y_j),$
                       $D(i, j - 1) + C(y_j, x_i, x_{i-1}))$
10: return $D(m, n)$
```

2.9 Other approaches

The elastic distance measures discussed in Sect. 2 are the current state of the art for TSC in the time domain. However, a number of algorithms have recently been proposed that utilise alternative data representations. These operate by either explicitly transforming the data and then constructing a classifier, or by embedding the transformation within the classification algorithm. Our primary experimental objective is comparing classifiers based on elastic distance measures. However, we believe that no one representation or approach will dominate all others, so for the sake of completeness and context we briefly review the most prominent of the recent approaches. In Sect. 6.2 we compare, where possible, our results to those published in the papers referenced below.

2.9.1 Shapelet representation

One alternative to time domain classifiers is time series shapelets (Ye and Keogh 2009; Rakthanmanon and Keogh 2013; Hills et al. 2013). Shapelets are time series subsequences that are discriminatory of class membership. They allow for the detection of phase-independent localised similarity between series within the same class. The original shapelets algorithm by Ye and Keogh (2009) uses a shapelet as the splitting criterion for a decision tree. Hills et al. (2013) propose a shapelet transformation that separates the shapelet discovery from the classifier by finding the top k shapelets on a single run (in contrast to the decision tree, which searches for the best shapelet at each node). The shapelets are used to transform the data, where each attribute in the new dataset represents the distance of a series to one of the shapelets. They demonstrate that the ability to use more sophisticated classifiers in conjunction with shapelets significantly reduces classification error. Both (Ye and Keogh 2009; Hills et al. 2013) find shapelets through enumeration. Rakthanmanon and Keogh (2013) present an approximate shapelet finding approach based on discretisation and random projection that significantly speeds up the shapelet discovery process.

2.9.2 Interval features

A further approach at finding localised discriminatory features is based on summary statistics calculated over different width intervals of a series. For a series of length m, there are $m(m-1)/2$ possible contiguous intervals. Rodriguez et al. (2005) calculate binary features over these intervals, based on threshold rules on the interval mean and standard deviation. A subset of features are created through a boosting scheme, then transformed into an interval feature space that forms the training data for a support vector machine.

Deng et al. (2013) calculate three statistics on each of the possible features. They calculate the mean, standard deviation, and slope, and use these features to construct classifiers. Rather than generate the entire new feature space of $3m(m-1)/2$ attributes, they use a random forest classifier, with each member of the ensemble assigned a random subset of features from the interval feature space.

2.9.3 Pattern frequency features

In information retrieval, the bag-of-words approach of estimating word frequencies and ignoring location is very common. Two recent papers have adopted this approach for time series classification. The idea is to estimate the frequency of occurrence of localised characteristics of the series, and then use these frequencies as new features for a classifier. Lin et al. (2012) propose a bag-of-patterns (BoP) approach that involves converting a time series into a discrete series using Symbolic Aggregate approximation (SAX) (Lin et al. 2007), creating a set of SAX words for each series through the application of a short sliding window, then using the frequency count of the words in a series as the new feature set.

Baydogan et al. (2013) describe a bag-of-features approach that combines interval and frequency features. The algorithm, called time series based on a bag-of-features representation (TSBF), involves separate feature creation and classification stages. The feature creation stage involves generating random intervals and then creating features representing the mean, variance and slope over the interval. The start and end point of the interval are also included as features in order to retain the possibility of detecting temporal similarity. There is then a further feature transform that involves supervised learning. The features of each interval form an instance, and each time series now represents a bag. A classifier is used to generate a class probability estimate for each instance. The probability estimates of all instances for a given time series (bag) are then discretised and a histogram for each possible class value is formed. The resulting concatenated histograms form the feature space for the training set of a classifier. A random forest classifier is used for the labelling and a random forest and support vector machine for the classification in Baydogan et al. (2013).

3 Data sets

We have collected 75 data sets, the names of which are shown in Table 1. 46 of these are available from the UCR repository (Keogh et al. 2011), 24 were used in other published work (Bagnall 2012; Hills et al. 2013; Rakthanmanon and Keogh 2013) and 5 are new data sets we present for the first time. Further information and the data sets we have permission to circulate are available from (Lines and Bagnall 2014). We have removed the dataset ECG200 from all experiments, because an error in data processing means that it can be perfectly classified with a single rule on the sum of squared values for each series [see Bagnall et al. (2012)] for further details).

Table 1 Data sets grouped by problem type. The actual file names are in a string array in the supporting code

Image Outline Classification			
DistPhalanxAge	DistPhalanxOutline	DistPhalanxTW	FaceAll
MidPhalanxAge	idPhalanxOutline	MidPhalanxTW	FaceFour
ProxPhalanxAge	ProxPhalanxOutline	ProxPhalanxTW	WordSynonyms
OSULeaf	Phalanges	yoga	ShapesAll
SwedishLeaf	MedicalImages	Symbols	Adiac
ArrowHead	BeetleFly	BirdChicken	DiatomSize s
FacesUCR	fiftywords	fish	Herring
Motion Classification			
CricketX	CricketY	CricketZ	GunPoint
UWaveX	UWaveY	UWaveZ	UWaveAll
Haptics	InlineSkate	ToeSeg1	ToeSeg2
Sensor Reading Classification			
Beef	Car	Chlorine	CinCECG
Coffee	Computers	FordA	FordB
ItalyPower	LargeKitchen	Lightning2	Lightning7
StarLightCurves	Trace	TwoLeadECG	wafer
RefrigerationDevices	MoteStrain	Earthquakes	ECGFiveDays
ElectricDevices	SonyRobot1	SonyRobot2	OliveOil
Plane	Screen	SmallKitchen	ECGThorax1
	ECGThorax2		
Simulated Classification Problems			
ARSim	CBF	SyntheticControl	ShapeletSim
	TwoPatterns	MALLAT	

Furthermore, as also recommended in Bagnall et al. (2012), we have normalised the datasets Coffee, Olive Oil and Beef.

We have grouped the problems into categories to help aid interpretation. The group of sensor readings forms the largest category with 29 data sets. If we had more data sets it would be sensible to split the sensor categories into subtypes, such as human sensors and spectrographs. However, at this point such sub-typing would lead to groups that are too small. Image outline classification is the second largest category, with 28 problem sets. Many of the image outline problems, such as *BeetleFly*, are not rotationally aligned, and the expectation would be that classifiers in the time domain will not necessarily perform well with these data. The group of 12 motion problems contains data taken from motion capture devices attached to human subjects. The final category is simulated data sets.

3.1 Electricity datasets

We provide five new classification problems that focus on domestic electricity consumption within the United Kingdom. These problems were taken from data recorded as part of government sponsored study (Trust 2012). The intention was to collect behavioural data about how consumers use electricity within the home to help reduce the UK's carbon footprint. The aim is to reduce CO_2 emissions 80 % by 2050, and it is clear that reductions in household consumption will be crucial to meet this goal. Much recent attention has been placed on how to reduce consumer consumption without a perceived reduction to quality of life, and one attempt to do this by the UK government is to install smart meters with real-time displays into every household. At a cost of £11.1 bn, the project aims to alert consumers of their current consumption with the primary objective to show consumers their real-time spending, with the hope that this insight will cause them to reduce unnecessary consumption to save money. However, this project will also create vast streams of data from the 30 million households that will be monitored, so it is imperative that further information is extracted from the data to support other projects, such as creating smart energy grids and managing the creation of renewable energy.

The datasets that we introduce support the goal of promoting further understanding of domestic electricity consumption by classifying appliances according to their electricity usage profiles. The data contains readings from 251 households, sampled in two-minute intervals over a month. When designing the classification problems we took our previous work in Lines et al. (2011) into account. We originally classified 10 device types according to daily and weekly profiles sampled over 15-minute intervals. This work influenced the creation of the new datasets in two regards. Firstly, we found that if consumption data from a specific device in a given household were included in both the training and testing data, bias was introduced into classification results as decisions were made by matching the specific device, rather than classifying the class of the devices. Secondly, we noted it was difficult to differentiate between devices with similar purposes and behavioural patterns. Therefore, for the datasets we create two distinct types of problem: problems with similar usage patterns (Refrigeration, Computers, Screen) and problems with dissimilar usage patterns (Small Kitchen and

Table 2 The five new TSC problems with class values

Problem	Class labels
Small kitchen	Kettle, microwave, toaster
Large kitchen	Dishwasher, tumble dryer, washing machine
Refrigeration	Fridge/freezer, refrigerator, upright freezer
Computers	Desktop, laptop
Screen	CRT TV, LCD TV, computer monitor

Large Kitchen). The aim is that problems with dissimilar usage patterns should be well suited to time-domain classification, whilst those with similar consumption patterns should be much harder. The five problems we form are summarised in Table 2. Further information on these data sets can be found at Lines and Bagnall (2014).

3.2 Hand/bone outlines datasets

A project on automated bone ageing gave rise to 10 image outline classification problems (Davis et al. 2012). Algorithms to automatically extract the hand outlines and then the outlines of three bones of the middle finger (proximal, middle and distal phalanges) were applied to over 1,300 images, and three human evaluators labelled the output of the image outlining as correct or incorrect. This generated three classification problems (*DistPhalanxOutline, MidPhalanxOutline, ProxPhalanxOutline*). The next stage of the project was to use the outlines to predict information about the subjects age. The three problems *DistPhalanxAge, MidPhalanxAge and ProxPhalanxAge* involve using the outline of one of the phalanges to predict whether the subject is one of three age groups: 0–6 years old, 7–12 years old and 13–19 years old. Note that these problems are aligned by subject, and hence can be treated as a multi dimensional TSC problem. Data file *Phalanges* contains the concatenation of all three problems. Bone age estimation is usually performed by an expert with an algorithm called Tanner-Whitehouse (2001). This involves scoring each bone into one of seven categories based on the stage of development. The final three bone image classification problems, *DistPhalanxTW, MidPhalanxTW and ProxPhalanxTW*, involve predicting the Tanner-Whitehouse score (as labelled by a human expert) from the outline.

4 Ensemble schemes

An ensemble of classifiers is a set of base classifiers whose individual decisions are combined through some process of fusion to classify new examples. One key concept in ensemble design is the requirement to inject diversity into the ensemble. Broadly speaking, diversity can be achieved in an ensemble by: employing different classification algorithms to train each base classifier to form a heterogeneous ensemble; changing the training data for each base classifier through a sampling scheme or by directed replication of instances (for example, Bagging); selecting different attributes to train each classifier, often randomly; or modifying each classifier internally, either through re-weighting the training data or through inherent randomization (for example, AdaBoost).

Ensembles have been used in time series data mining (Deng et al. 2013; Bagnall et al. 2012). Deng et al. (2013) propose a random forest (Breiman 2001) built on summary statistics of intervals. They compare against a random forest built on the raw data, 1-NN with Euclidean distance and 1-NN with DTW. Their results suggest that the transforms they propose significantly improve the random forest classifier. Buza (2011) proposes an ensemble that combines alternative elastic and inelastic distance measures and compares performance to DTW and stacking (an alternative form of ensemble) support vector machines. Our approach is more transparent. We construct each of the classifiers independently, then assess alternative voting schemes for using these predictions.

We have experimented with four separate, and very simple, weighting schemes. The first, *Equal*, gives equal weight to the predictions of each classifier. Clearly the fastest to train, it nevertheless will be deceived on problems where the classifier is inaccurate. The other techniques are based on the cross validation accuracy on the training set. *Best* assigns a weight of 1 to the transform with the highest CV accuracy and zero to the others. *Equal* is at one extreme (ignoring the information from training), whereas *Best* is at the other. If the CV accuracy is a good estimator (i.e. unbiased and low variance) of the true accuracy, then *Best* will be the optimal scheme. However, with small data set sizes and complex underlying models, this approach may be brittle. In effect we are putting too much faith in the CV accuracy. To counteract this, we also use a method *Proportional*, which assigns the weight as the CV accuracy, normalised over the number of transformations, and *Significant*. *Significant* is a combination approach of *Best* and *Proportional*. It only includes the classifiers which are not significantly worse than the classifier with the highest CV accuracy. We start by finding the classifier with the highest CV accuracy (ties are split randomly). We then test significance using McNemar's test, which is based on the counts of coincidence of correct and incorrect predictions. If we reject the null hypothesis of no difference to the best classifier, the weight of classifier in question is set to zero. If we cannot reject the null, it is set to the appropriate CV accuracy. This approach is meant to remove clearly inferior transforms whilst allowing for smaller inaccuracies in the power of the CV accuracy to estimate the true accuracy.

5 Comparison of elastic measures

We conducted our classification experiments with WEKA (Hall et al. 2009) source code adapted for time series classification, using the 75 data sets described in Sect. 3.

All datasets are split into a training and testing set, and all parameter optimisation is conducted on the training set only. We use a train/test split for several reasons. Firstly, it is common practice to do so with the UCR datasets (for example, the papers proposing WDTW (Jeong et al. 2011) and TWE (Marteau 2009) both use a test train split). Secondly, some of the data sets are designed so the train/test split removes bias (for example, see Sect. 3.1). Combining to perform a cross validation would reintroduce the bias.

Our data sets have been randomly selected, in the sense that we are not attempting to find data sets more suitable for one particular algorithm. However, as Table 1 demonstrates, there is a bias in areas of application we are considering. There are, for example, no problems from econometrics or finance. Because of this, we also present results split by problem type where appropriate.

In proposing WDTW, Jeong *et al.* present results on 20 UCR data sets and claim that *"our proposed distance measures, WDTW and WDDTW, clearly outperform standard DTW, DDTW and LCSS"* (Jeong et al. 2011). In Marteau (2009), TWE is evaluated on the same 20 data sets, and it is claimed *"the TWED distance is effective ... since it exhibits, on the average, the lowest error rates for the testing data"*. As is common in much of the time series classification literature, the support for these algorithms is based on an informal consideration of the number of wins or average accuracy rather than tests of significance. Our objective is to assess if there is any significant difference in accuracy (or equivalently, error) between eleven related distance measures: Euclidean distance (ED), dynamic time warping with full window (DTW), derivative DTW with full window (DDTW), DTW and DDTW with window size set through cross validation (DTWCV and DDTWCV), weighted DTW and DDTW (WDTW and WDDTW), LCSS, ERP, TWE, and the MSM. All parameter optimisation is conducted on the training set through cross validation, with each algorithm allowed 100 model selection evaluations. In machine learning, the generally accepted statistic for comparing k classifiers over n data sets is a non-parametric form of Analysis of Variance based on ranks, described in Demšar (2006). The null hypothesis is that the average rank of k classifiers on n data sets is the same, against the alternative that at least one classifier's mean rank is different. So, given an n by k matrix M of error rates, the first stage is to evaluate the n by k rank matrix R, where r_{ij} is the rank of the jth classifier on the ith data set. The ranks of those classifiers that have equal error are averaged. The average rank of each classifier is then denoted $\bar{r}_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}$. Under the null hypothesis of the ranks being equal for all classifiers, the Friedman statistic Q,

$$Q = \frac{12n}{k(k+1)} \cdot \left[\sum_{j=1}^{k} \bar{r}_j^2 - \frac{k(k+1)^2}{4} \right]$$

can be approximated by a a Chi-squared distribution with $(k-1)$ degrees of freedom. However, Demšar (2006) identifies that previous work has shown that this approximation to the distribution of Q is conservative, and proposed using

$$F = \frac{(n-1)Q}{n(k-1)-Q}$$

which, under the null hypothesis, follows an F distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom. If the result of the test is to reject the null hypothesis, Demšar recommends grouping classifiers into *cliques*, within which there is no significant difference in rank. This allows the average ranks and groups of not signif-

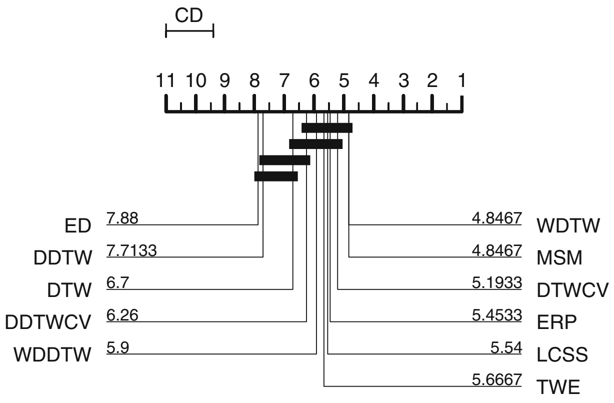

Fig. 4 The average ranks for eleven 1-NN classifiers on the 75 data sets described in Sect. 3. The critical difference for $\alpha = 5\,\%$ is 1.6129

icantly different classifiers to be plotted on an order line in a graph referred to as a critical difference diagram.

Figure 4 shows the critical difference diagram for eleven 1-NN classifiers on the 75 data sets. We observe that, firstly, there is no one classifier that significantly outperforms the others. There are four cliques within which no significant difference is observed. The top clique contains all but ED, DDTW and DTW. There is no significant difference between any of the elastic measures. The conclusions we can draw from these results is that, firstly, it is easy to outperform Euclidean distance, secondly, setting window size through cross validation improves the ranks of both DTW and DDTW (an observation first made in Ratanamahatana and Keogh (2005)) and thirdly, introducing weights (WDTW) also significantly improves full window DTW.

These results do not lend any weight to one elastic algorithm over the others. Given the standard practice is to use DTWCV (Batista et al. 2011), the obvious question is, do our experiments indicate that there is any reason to use any of the other classifiers?

5.1 Time complexity comparison

In addition to comparing error rates across our 75 datasets, we can also evaluate the time complexities of the individual distance measures. There could be a case for dismissing distance measures if they have relatively poor time complexities and do not improve on the error rates of other measures. However for the measures that we consider, each has a basic time complexity of $O(n^2)$ (with the exception of the Euclidean Distance which is $O(n)$), but this is confounded slightly when considering parameter options. For example, constraining a DTW search with a warping window reduces the time complexity to $O(nr)$, where r is the size of the warping that is allowed by the window. In general however, there is little to choose from when comparing the runtime of the different measures. The graph presented in Fig. 5 demonstrates the difference in time taken between the measures when calculating the distance between two series from each of the 75 datasets. For the purpose of presentation, each of the 100 parameter

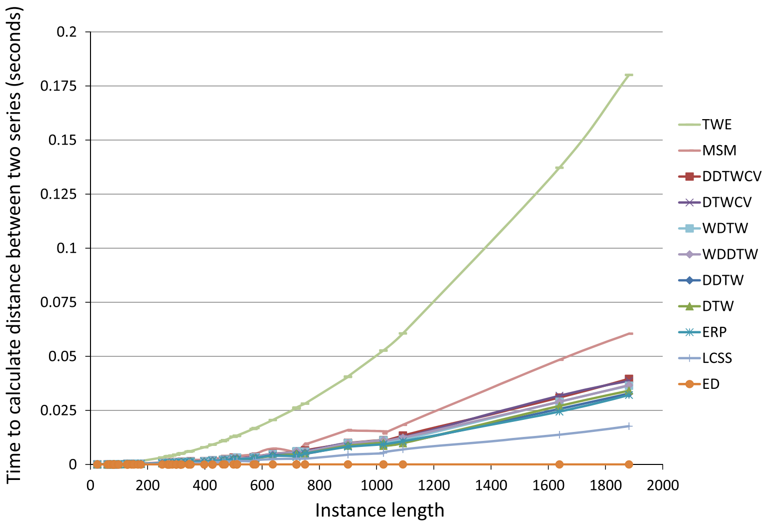

Fig. 5 A comparison of the runtime to compute the distance of two series with each distance measure for each of the 75 datasets. The times were calculated by running each parameter option 100 times and taking the median for each option, then taking the median of the 100 options for each dataset

selections for each distance measure were run 100 times and the median was taken for each, and then the median of the resulting 100 parameters options was selected for presentation. It is interesting to note that in Fig. 5, TWE does not scale as well as the other $O(n^2)$ measures. However, since the time complexity is still $O(n^2)$, we choose not to omit any measures due to their time complexities.

It should be noted that we have used the standard implementations of each type of distance measure in our work. It may be possible to speed up certain components of measures (for example, using parallelisation or lower bounding). However, we wish to keep our message clear so use the standard implementations of each measure without implementing any such speedups.

6 Ensembles of elastic distance measures

One reason for not using DTWCV would be if we could determine on the training data which alternative classifier is the most appropriate. We can assess the *a priori* usefulness of alternative classifiers with a type of plot introduced in Batista et al. (2013). The basic principle is that the ratio of the training accuracy of two classifiers (generated through cross validation) should give an indication to the outcome for the test data. However, if the cross validation accuracy is biased or subject to high variance, then often the ratio will be misleading. The plot of training accuracy ratio vs. testing accuracy ratio give a continuous form of contingency table for assessing the usefulness of the training accuracy. If the ratio on training data and testing data are both greater than one then the case is true positive (we predict a gain for one algorithm and also observe a gain). If both ratios are less than one, the problem is a true negative (we

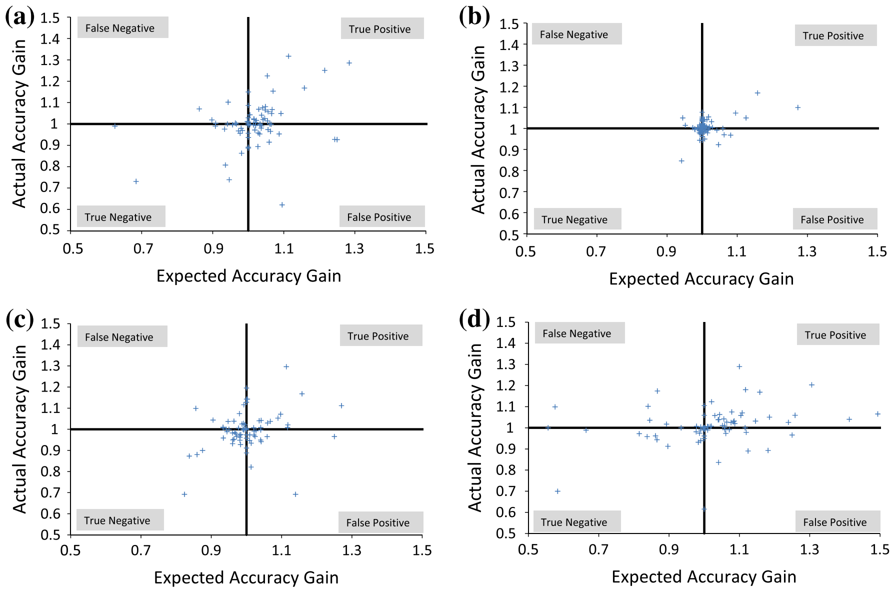

Fig. 6 Texas sharpshooter plots of DTWCV vs. **a** LCSS, **b** WDTW, **c** TWE and **d** MSM. The *points* on the plots represents results for each of the 75 datasets, where expected accuracy gain is calculated as the ratio of the each of measure vs. DTWCV on the training data and the actual accuracy gain is calculated as the ratio between the measures and DTWCV on the test data. If training CV accuracies of the measures offer a good a priori indication of test performance, we would expect there to be few false results. However, there is no clear trend of true positives or true negatives with any of the four measures, so simply relying on the training CV accuracies to determine which measure will be most effective in advance is not a reliable strategy

predict a loss and also observe a loss). Otherwise, we have an undesirable outcome. If the data sets are evenly spread between the four quadrants, then Batista et al. observe that we have a situation analogous to the Texas sharpshooter fallacy (which comes from a joke about a Texan who fires shots at the side of a barn, then paints a target centered on the biggest cluster of hits and claims to be a sharpshooter). Figure 6 shows the graphs for DTWCV against LCSS, WDTW, TWE, and MSM.

The number of data sets in true positive or true negative are 38, 31, 35 and 49 respectively. This means that using the cross validation accuracy to determine which classifier is going to be better on the test data is worse than guessing in three out of four cases. If there is a difference between the classifiers, then there is too much bias and/or variance in the training accuracy to detect it.

A second reason for using WDTW, LCSS, ERP, TWE, or MSM, rather than DTWCV, would be if they performed better on a particular problem domain. Table 3 shows the average ranks of these six distance measures on the different problem types. LCSS and MSM perform well on the image outlines, WDTW is best on motion, and ERP on sensor. The two DTW approaches rank highly on motion and sensor, but relatively poorly on image outlines. LCSS is in many ways closer to a Shapelet approach (Rakthanmanon and Keogh 2013) than DTW, and this result suggests that subsequence matching techniques such as LCSS and Shapelets may be better for image

Table 3 Average ranks of six classifiers across the 75 datasets split by problem type (as originally defined in Table 1)

Type	DTWCV	WDTW	LCSS	MSM	TWE	ERP
IMAGE	3.93	3.91	2.89	2.79	3.48	4.00
MOTION	3.29	2.67	4.04	3.17	3.88	3.96
SENSOR	3.22	3.16	4.34	3.43	3.78	3.07
SIM	4.08	2.08	3.67	3.50	3.92	3.75
Overall	3.57	3.27	3.70	3.15	3.69	3.61

Table 4 Number of data sets where the individual classifiers are significantly different from each other

	DTWCV	WDTW	LCSS	TWE	ERP	MSM
DTWCV	0	1	30	22	14	23
WDTW	1	0	26	21	12	22
LCSS	30	26	0	24	23	18
TWE	22	21	24	0	19	17
ERP	14	12	23	19	0	20
MSM	23	22	18	17	20	0

outline classification. TWE is worst on the simulated set (the least interesting category) but is mid ranked on the other categories. This may be a reflection of the hybrid nature of the algorithm.

Whilst the problem type may give some indication as to the best measure to use, it is far from conclusive. A third possible use for the alternative measures is within an ensemble. The key design criteria for a successful ensemble is that the components produce diverse predictions. If the classifiers are approximately equal in accuracy and yet divergent in predictions, then it is possible to reduce the variance in prediction and hence produce an overall more accurate classifier. Put more simply, if they are getting a different subset of cases correct, then there is the possibility of combining their predictions to increase the overall number correct.

To measure divergence in predictions, we perform a pairwise McNemar's test between each of our classifiers, and count the number of data sets where the null hypothesis that the probability of misclassification is the same either way can be rejected. Table 4 gives the number of significant differences (at the 5 % level) between pairs of distance functions. There is only a single data set where WDTW is significantly different to DTWCV, but both DTWCV and WDTW differ on a high proportion of the data sets when compared to LCSS and TWE. Furthermore, TWE is different on over twenty data sets in relation to all of the other classifiers.

Clearly, whilst the overall performances are not significantly different, there is significant diversity in predictions on many of the data sets. Hence, we test whether a significantly more accurate classifier can be obtained by combining the predictions of the alternative elastic distance measure with the ensemble techniques described in Sect. 4. Figure 7 shows the average ranks and critical differences for the top six individual classifiers and the four ensembles. We have omitted the results of the worst performing constituent classifiers for clarity. The full results are available on the associated website (Lines and Bagnall 2014).

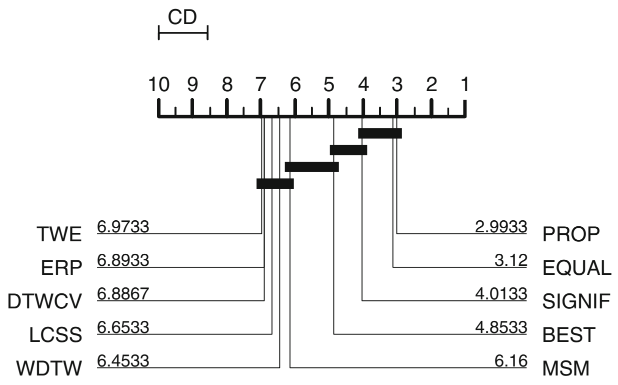

Fig. 7 The average ranks for the six highest ranked individual classifiers and four ensemble techniques across the 75 datasets

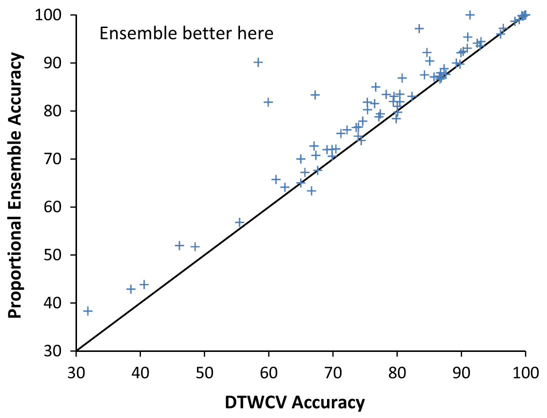

Fig. 8 Scatter plot of test accuracies of a single DTWCV classifier against the proportional ensemble

The three ensemble techniques *Equal*, *Significant* and *Proportional* are all significantly better than all of the individual classifiers. The ensemble *Best* does not actually give significant improvement over MSM. This reiterates the danger of placing too much emphasis on training cross validation measures. Our attempt to ensemble using an inclusion threshold (*Significant*) does not provide any improvement over weighting proportional to CV accuracy.

The improvement in performance of the ensembles over the single classifiers is dramatic. The average rank of the proportional ensemble is 3.17 higher than the best performing single classifier (MSM). This difference gives a P-value of 0.002. The proportional ensemble is better than all of DTWCV, WDTW, LCSS, ERP, TWE and MSM on 46 of the 75 data sets, and better than at least two of these six classifiers on all data.

We believe the proportional ensemble is the first classifier to be significantly more accurate than DTWCV on TSC problems. Figure 8 demonstrates this improvement through a scatter plot of accuracies of DTWCV against the ensemble. Plots against LCSS, WDTW, MSM, and TWE exhibit a very similar pattern.

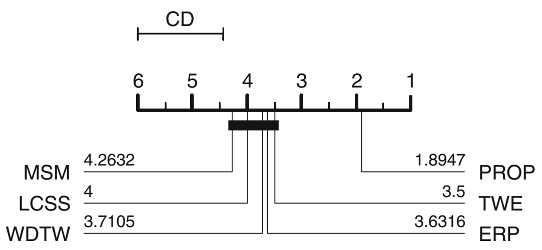

Fig. 9 Ranks of four classifiers on the original 19 UCR data sets (omitting ECG)

Finally, to counter any accusations of implementation bias or data set cherry picking, we compare our proportional ensemble results on the original 19 UCR data sets used in the evaluation of Marteau (2009); Jeong et al. (2011) with the results reported in the papers. Figure 9 shows the resulting critical difference diagram. Firstly, the author's own results show no critical difference between DTWCV and WDTW, TWE or MSM. This is despite the fact that the results reported for WDTW include allowing the algorithm half the test data for model selection (an advantage they did not give to DTWCV). Secondly, the ensemble results are significantly better.

There are numerous metrics for assessing new algorithms for TSC. While we believe that accuracy is the most important of these, it is also important to consider the time complexity of new approaches. The ensemble is clearly more time consuming than DTWCV and the other single measures in practice as the ensemble is a combination of 11 individual measures. However, no constituent measure has a time complexity worse than $O(n^2)$, which results in the ensemble also having $O(n^2)$ time complexity. With this in mind, it should be noted that the design of the ensemble lends itself well to modern computing architectures. As the ensemble is made up of distinct constituent classifiers, there is ample potential for running constituents as subroutines in a parallel architecture to accelerate classification performance. As previously discussed in Sect. 5.1 however, we have not implemented any such speedups in this work to allow us to maintain focus on the classification accuracy of the ensemble against DTWCV and the other elastic measures.

6.1 Internal classifier contribution

The results of the proportional ensemble leads us to a question: is there a subset of classifiers that, when coupled with DTWCV, is responsible for significantly improving performance when compared to DTWCV alone, or is the performance gain due to the weighted contribution of all constituent classifiers? We investigate this by creating an incremental version of the proportional ensemble. In the first iteration, the ensemble contains the DTWCV classifier only. With each subsequent iteration, we add the best ranked classifier from our experiments that is not already included, and classify each of the 75 datasets using the proportional weighting scheme at each step. We repeat this process until each of the 11 constituent classifiers are included in the ensemble. The average error rates across all datasets at each iteration are shown in Fig. 10.

The error rates in Fig. 10 appear to steadily decrease with the inclusion of additional classifiers, rather than substantially changing with the inclusion of a specific classifier.

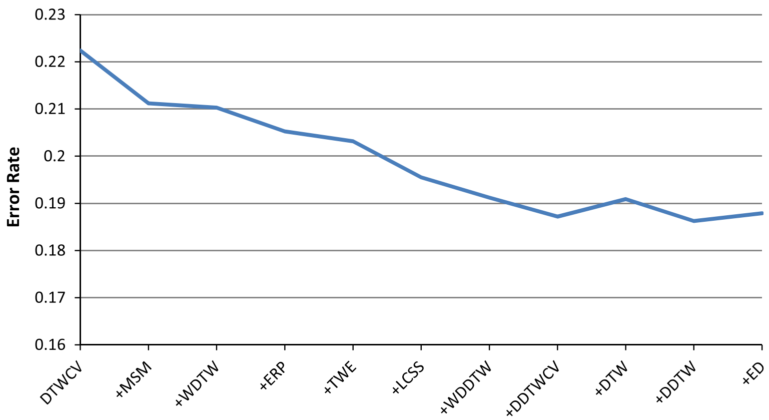

Fig. 10 The error rates of the proportional ensemble scheme when adding classifiers incrementally. The ensemble initially contains DTWCV only, and at each iteration, the best-ranked classifier that is not currently included is added

This trend tails off towards the end of the experiment however, and the error rates appear to plateau after the inclusion of DDTWCV. This can be explained because the subsequent classifiers, DTW, DDTW, and ED, should already be represented within the ensemble on problems where they are an effective solution. For example, if we take a problem where DTW with a full window is the optimal solution, the DTWCV within the ensemble should also have a window set to 100% after carrying out CV. The consequence of this is that in cases where the DTW classifier is optimal, the DTWCV should be identical (assuming CV correctly detects that a full window is optimal). The equivalent theory is also true for DDTW with DDTWCV, and ED with DTWCV and a window size of 1 %.

These results suggest that the performance of the ensemble is dependent on the weighted votes of the majority of the constituent classifiers, rather than a small subset. There is a case for removing the simpler, constrained, classifiers that do not have values set through CV as they do not appear to positively influence classification. However, they do not reduce accuracy either, so for clarity we continue to consider them within the ensemble to keep results consistent.

6.2 Comparison of the proportional ensemble to other approaches

There are currently 47 UCR data sets available, and a subset of these have been used in all recent research on TSC. All our comparison results are available on the accompanying website. In all of the analysis below, we discount ECG200. Two are not in the general release (car and plane), so the greatest number of results we compare against is 44. We test the difference between pairs of algorithms using a Wilcoxon signed rank test with α set to 1 %. A full list of the results is presented in Tables 5 and 6.

Fast shapelets and logical shapelets (Ye and Keogh 2009; Rakthanmanon and Keogh 2013) were evaluated on 33 UCR datasets. One is ambiguous (named Cricket, when

Table 5 The full results of the state of the art in TSC on the UCR datasets (part one of two)

Dataset	ED	DTWCV	TSF	TSBF	CID	PROP
Adiac	0.389	0.391	0.261	**0.245**	0.379	0.353
Beef	0.467	0.467	0.3	0.287	0.467	0.367
Car	0.267	0.233	–	–	–	**0.167**
CBF	0.148	0.004	0.039	0.009	**0.001**	0.002
ChlorineConcentration	0.35	0.35	0.26	0.336	0.351	0.36
CinC_ECG_torso	0.103	0.07	0.069	0.262	0.054	0.062
Coffee	0.25	0.179	0.071	0.004	0.179	**0**
Cricket_X	0.426	0.236	0.287	0.278	0.249	**0.203**
Cricket_Y	0.356	0.197	0.2	0.259	0.197	**0.156**
Cricket_Z	0.38	0.18	0.239	0.263	0.205	**0.156**
DiatomSizeRed.	0.065	0.065	0.101	0.126	0.065	**0.059**
ECGFiveDays	0.203	0.203	0.07	0.183	0.218	0.178
FaceAll	0.286	0.192	0.231	0.234	**0.144**	0.152
FaceFour	0.216	0.114	0.034	0.051	0.125	0.091
FacesUCR	0.231	0.088	0.109	0.09	0.102	**0.063**
fiftywords	0.369	0.242	0.277	0.209	0.226	**0.18**
fish	0.217	0.16	0.154	0.08	0.154	0.034
GunPoint	0.087	0.087	**0.047**	0.011	0.073	0.007
Haptics	0.63	0.588	0.565	**0.488**	0.571	0.584
InlineSkate	0.658	0.613	0.675	0.603	0.586	0.567
ItalyPowerDemand	0.045	0.045	**0.033**	0.096	0.044	0.039
Lightning2	0.246	0.131	0.18	0.257	0.131	**0.115**
Lightning7	0.425	0.288	0.263	0.262	0.26	**0.233**
MALLAT	0.086	0.086	0.072	0.037	0.075	0.05
MedicalImages	0.316	0.253	**0.232**	0.269	0.258	0.245
MoteStrain	0.121	0.134	0.118	0.135	0.205	0.114
NonInvasive.Thorax1	0.171	0.185	**0.103**	0.138	–	0.178
NonInvasive.Thorax2	0.12	0.129	**0.094**	0.13	–	0.112
OliveOil	0.133	0.167	0.1	**0.09**	0.167	0.133
OSULeaf	0.483	0.384	0.426	0.329	0.372	**0.194**
Plane	0.038	**0**	–	–	–	**0**
SonyAIBORobot.	0.141	0.141	0.235	0.175	0.185	0.293
SonyAIBORobot.II	0.305	0.305	0.177	0.196	0.123	0.124
StarLightCurves	0.151	0.095	0.036	**0.022**	0.066	0.079
SwedishLeaf	0.213	0.157	0.109	**0.075**	0.117	0.085
Symbols	0.1	0.062	0.121	**0.034**	0.059	0.049
SyntheticControl	0.12	0.017	0.023	**0.008**	0.027	0.01
Trace	0.24	0.01	**0**	0.02	0.01	0.01
TwoLeadECG	0.253	0.132	0.112	**0.046**	0.138	0.067
TwoPatterns	0.09	0.0015	0.053	0.001	0.004	**0**

Table 5 continued

Dataset	ED	DTWCV	TSF	TSBF	CID	PROP
UWaveGesture._X	0.261	0.227	0.213	**0.164**	0.211	0.199
UWaveGesture._Y	0.338	0.301	0.288	**0.249**	0.278	0.283
UWaveGesture_Z	0.35	0.322	0.267	**0.217**	0.293	0.29
wafer	0.005	0.005	0.047	0.004	0.006	0.003
WordSynonyms	0.382	0.252	0.381	0.302	0.243	0.226
yoga	0.17	0.155	0.157	0.149	0.156	**0.121**

The results are split into two parts for readability; part one includes results for Euclidean Distance (ED), DTW with window set through CV (DTWCV), Time Series Forests (TSF) (Deng et al. 2013), Time Series Bag of Features (TSBF) (Baydogan et al. 2013), Complexity-Invariant Distance (CID) (Batista et al. 2013), and the Proportional Elastic Ensemble (PROP). Dashes indicate where results were not available. The results are continued in Table 6, and the best result on each dataset across both tables is shown in bold

Table 6 The full results of the state of the art in TSC on the UCR datasets (part two of two)

Dataset	BoP	LS	FS	ST	FUSION	TE	PROP
Adiac	0.432	0.414	0.514	0.486	0.352	0.358	0.353
Beef	0.433	0.433	0.447	**0.167**	–	0.4	0.367
Car	–	–	–	–	0.283	–	**0.167**
CBF	0.013	0.114	0.053	**0.001**	–	0.171	0.002
ChlorineConcentration	**0.036**	0.382	0.417	0.333	0.473	–	0.36
CinC_ECG_torso	–	0.301	0.174	0.137	**0.043**	–	0.062
Coffee	–	0.036	0.068	**0**	–	0.214	**0**
Cricket_X	–	–	–	–	–	–	**0.203**
Cricket_Y	–	–	–	–	–	–	**0.156**
Cricket_Z	–	–	–	–	–	–	**0.156**
DiatomSizeRed.	–	0.199	0.117	0.127	–	–	**0.059**
ECGFiveDays	–	0.006	0.004	**0**	–	–	0.178
FaceAll	0.219	0.341	0.411	0.254	–	0.281	0.152
FaceFour	**0.023**	0.511	0.09	0.102	0.116	0.148	0.091
FacesUCR	–	0.338	0.328	–	0.077	–	**0.063**
fiftywords	0.466	–	–	–	0.224	0.352	**0.18**
fish	0.074	0.223	0.197	**0.029**	–	0.194	0.034
GunPoint	0.093	0.107	0.061	0.013	–	0.053	0.007
Haptics	–	–	–	–	0.587	–	0.584
InlineSkate	–	–	–	–	**0.462**	–	0.567
ItalyPowerDemand	–	0.064	0.095	0.052	–	–	0.039
Lightning2	0.164	0.574	0.295	0.393	0.206	0.23	**0.115**
Lightning7	0.466	0.452	0.403	**0.233**	0.251	0.301	**0.233**
MALLAT	–	0.344	**0.033**	0.074	0.1	–	0.05
MedicalImages	–	0.413	0.433	0.393	0.234	–	0.245
MoteStrain	–	0.168	0.217	0.111	**0.07**	–	0.114

Table 6 continued

Dataset	BoP	LS	FS	ST	FUSION	TE	PROP
Plane	–	–	–	–	–	–	**0**
SonyAIBORobot.	–	0.14	0.314	**0.105**	–	–	0.293
SonyAIBORobot.II	–	0.154	0.215	**0.109**	–	–	0.124
StarLightCurves	–	–	–	–	0.15	–	0.079
SwedishLeaf	0.198	0.187	0.269	0.118	0.115	0.157	0.085
Symbols	–	0.357	0.068	0.118	–	–	0.049
SyntheticControl	0.037	0.53	0.081	0.02	–	0.083	0.01
Trace	**0**	**0**	0.002	0.02	–	0.2	0.01
TwoLeadECG	–	0.144	0.09	–	–	–	0.067
TwoPatterns	0.129	0.461	0.113	0.082	–	0.165	**0**
UWaveGesture._X	–	–	–	–	-	–	0.199
UWaveGesture._Y	–	–	–	–	–	–	0.283
UWaveGesture_Z	–	–	–	–	–	–	0.29
wafer	0.003	**0.001**	0.004	**0.001**	–	0.002	0.003
WordSynonyms	–	–	–	–	**0.214**	–	0.226
yoga	0.17	0.26	0.249	0.197	0.401	0.163	**0.121**

The results are split into two parts for readability. The first part is shown in Table 5, and part two includes results for Bag of Patterns (BoP) (Lin et al. 2012), Logical Shapelets (LS) and Fast Shapelets (FS) (Ye and Keogh 2009; Rakthanmanon and Keogh 2013), Shapelet Transform (ST) (Hills et al. 2013), Fusion Methods for TSC (FUSION) (Buza 2011), and the Transformation-Based Ensemble for TSC (TE) (Bagnall et al. 2012). The results of the Proportional Elastic Ensemble (PROP) are repeated for comparison. Dashes indicate where results were not available, and the best result on each dataset across both tables is shown in bold

there are in fact three Cricket files), hence we compare the elastic ensemble over 31 data sets. The proportional ensemble is more accurate than both shapelet tree algorithms on 26 of the 31 data sets, and worse on just 5. The difference is significant.

The results against the shapelet transform used in conjunction with a linear SVM (Hills et al. 2013) are less conclusive. The elastic ensemble wins on 17 of the 29 UCR data sets, ties on 3 and loses on 12. This difference is not significant.

Figure 11 shows the critical difference diagram for shapelets, DTWCV and the elastic ensemble on the 29 common data sets. Firstly, it demonstrates that the elastic ensemble is not significantly better than the shapelet transform, but is significantly better than the other classifiers. Secondly, the diagram reinforces the conclusion drawn in Hills et al. (2013) that the best way to use shapelets (at least in terms of accuracy) is to transform rather than embed in a decision tree. Finally, Fig. 11 shows that none of the shapelet algorithms are significantly better than DTW.

Rodriguez et al. (2005) evaluate their interval based SVM classifier on four UCR data sets: CBF; Two Patterns; Trace; and GunPoint. The ensemble performed better on three of these simple problems, with a very slightly worse accuracy on Trace.

Time Series Forests (TSF) (Deng et al. 2013) were evaluated on all 44 UCR data sets. The ensemble is better on 29, worse on 15. The difference is significant.

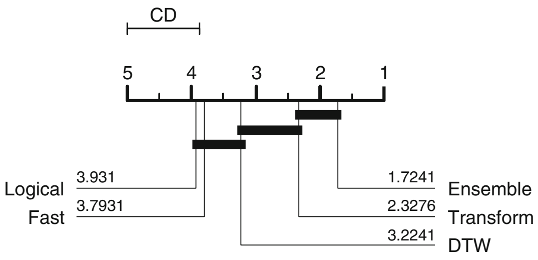

Fig. 11 The average ranks for three shapelet algorithms, DTWCV and Elastic Ensemble on 29 UCR datasets. The critical difference for $\alpha = 5\,\%$ is 1.1137

The BoP approach presented in Lin et al. (2012) is evaluated on 19 UCR data. The ensemble has significantly lower error than BoP. It wins on 15 data sets, is tied on 2 and is worse on 2 (FaceFour and Trace). The main support for the BoP algorithm is that it is much faster than DTW, but it is clear that this speed comes at the cost of accuracy.

The time series based on a bag-of-features representation (TSBF) described in Baydogan et al. (2013) is evaluated on 44 UCR data sets. The ensemble outperforms TSBF on 27 data sets and is worse on 17. The difference is significant.

The elastic ensemble is also significantly better than our previous ensemble described (Bagnall et al. 2012), winning on 21 out of 25 common data sets, which is a significant difference.

The fusion technique proposed in Buza (2011) was evaluated on 20 UCR data sets. The elastic ensemble was more accurate on 14 of these. The result is significant.

Finally, a recently proposed adjustment to Euclidean distance that compensates for the complexity of the series (Batista et al. 2011) was evaluated on 42 UCR data sets. The elastic ensemble was more accurate on 35 of these. The difference is significant.

7 Conclusions

We have conducted extensive experiments on the largest set of time series classification problems ever used in the literature (to the best of our knowledge) to demonstrate that, firstly, there is no significant difference between recently proposed elastic measures and secondly, that through ensembling we can produce a significantly more accurate classifier than DTWCV or any other elastic distance measure.

There are numerous metrics to assess new algorithms for TSC. However, accuracy is, in our opinion, the most important. We think that a new algorithm for TSC is only of interest to the data mining community if it can significantly outperform DTWCV and that to be of real interest it should outperform our proportional ensemble. We have made our code and data sets freely available to allow an objective comparison (Lines and Bagnall 2014).

We believe the proportional ensemble is the most accurate time series classifier yet to be proposed in the data mining literature. There have been numerous alternative classifiers for TSC that have published results on the UCR data [see, for example (Rakthanmanon and Keogh 2013; Lin et al. 2012; Deng et al. 2013)]. We have compared our results to all we can find in the literature, and the ensemble outperforms them all. There is not enough space to describe the analysis, but full details are available on the accompanying website (Lines and Bagnall 2014).

Adding alternative measures and achieving a deeper understanding of the importance and interaction of the components of the ensemble may lead to improved performance, and this is our next research objective.

References

Bagnall A (2012) Shapelet based time-series classification. http://www.uea.ac.uk/computing/machine-learning/shapelets

Bagnall A, Davis L, Hills J, Lines J (2012) Transformation based ensembles for time series classification. In: Proceedings of the 12th SDM

Batista G, Keogh E, Tataw O, de Souza V (2013) CID: an efficient complexity-invariant distance for time series. Data Mining and Knowledge Discovery online first

Batista G, Wang X, Keogh E (2011) A complexity-invariant distance measure for time series. In: Proceedings of the 11th SDM

Baydogan M, Runger G, Tuv E (2013) A bag-of-features framework to classify time series. IEEE Trans Pattern Anal Mach Intell 35(11):2796–2802

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Buza K (2011) Fusion methods for time-series classification. Ph.D. thesis, University of Hildesheim, Germany

Chen L, Ng R (2004) On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth international conference on Very large data bases, vol 30, pp 792–803. VLDB Endowment

Chen L, Özsu MT, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD international conference on management of data, pp 491–502. ACM

Davis L, Theobald BJ, Toms A, Bagnall A (2012) On the segmentation and classification of hand radiographs. Int J Neural Syst 22(5):12345-1–12345-2

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Deng H, Runger G, Tuv E, Vladimir M (2013) A time series forest for classification and feature extraction. Inf Sci 239:142–153

Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E (2008) Querying and mining of time series data: Experimental comparison of representations and distance measures. In: Proceedings of the 34th VLDB

Górecki T, Łuczak M (2013) Using derivatives in time series classification. Data Min Knowl Discov 26(2):310–331

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Hills J, Lines J, Baranauskas E, Mapp J, Bagnall A (2013) Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery online first

Hu B, Chen Y, Keogh E (2013) Time series classification under more realistic assumptions. In: Proceedings of the thirteenth SIAM conference on data mining (SDM). SIAM

Jeong Y, Jeong M, Omitaomu O (2011) Weighted dynamic time warping for time series classification. Pattern Recognit 44:2231–2240

Keogh E, Pazzani M (2011) Derivative dynamic time warping. In: Proceedings of the 1st SDM

Keogh E, Zhu Q, Hu B, Hao Y, Xi X, Wei L, Ratanamahatana C (2011) The UCR time series classification/clustering homepage. http://www.cs.ucr.edu/eamonn/time_series_data/

Lin J, Keogh E, Li W, Lonardi S (2007) Experiencing SAX: a novel symbolic representation of time series. Data Min Knowl Discov 15(2):107–144

Lin J, Khade R, Li Y (2012) Rotation-invariant similarity in time series using bag-of-patterns representation. J Intell Inf Syst 39(2):287–315

Lines J, Bagnall A (2014) Acompanying results, code and data. https://www.uea.ac.uk/computing/machine-learning/elastic-ensembles

Lines J, Bagnall A, Caiger-Smith P, Anderson S (2011) Intelligent data engineering and automated learning (IDEAL) 2011. Springer, New York, pp 403–412

Marteau PF (2009) Time warp edit distance with stiffness adjustment for time series matching. IEEE Trans Pattern Anal Mach Intell 31(2):306–318

Rakthanmanon T, Keogh E (2013) Fast-shapelets: A fast algorithm for discovering robust time series shapelets. In: Proceedings of the 13th SIAM international conference on data mining (SDM)

Ratanamahatana C, Keogh E (2005) Three myths about dynamic time warping data mining. In: Proceedings of the 5th SDM

Rodriguez J, Alonso C (2005) upport vector machines of interval-based features for time series classification. Knowl-Based Syst 18(4):171–178

Stefan A, Athitsos V, Das G (2012) The move-split-merge metric for time series. IEEE Trans Knowl Data Eng 25(6):1425–1438

Tanner J, Whitehouse R, Healy M, Goldstein H, Cameron N (2011) Assessment of skeletal maturity and prediction of adult height (TW3) method. Academic Press, New York

Trust ES (2012) Powering the Nation. Department for Environment, Food and Rural Affairs (DEFRA),

Ye L, Keogh E (2009) Time series shapelets: A new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD