The UCR Time Series Archive

Hoang Anh Dau[1], Anthony Bagnall[2], Kaveh Kamgar[1], Chin-Chia Michael Yeh[1], Yan Zhu[1], Shaghayegh Gharghabi[1], Chotirat Ann Ratanamahatana[3], Eamonn Keogh[1]

[1] *University of California, Riverside*

[2] *University of East Anglia*

[3] *Chulalongkorn University*

hdau001@ucr.edu, ajb@uea.ac.uk, kkamg001@ucr.edu, myeh003@ucr.edu, yzhu015@ucr.edu, sghar003@ucr.edu, chotirat.r@chula.ac.th, eamonn@cs.ucr.edu

Abstract— The UCR Time Series Archive - introduced in 2002, has become an important resource in the time series data mining community, with at least one thousand published papers making use of at least one dataset from the archive. The original incarnation of the archive had sixteen datasets but since that time, it has gone through periodic expansions. The last expansion took place in the summer of 2015 when the archive grew from 45 datasets to 85 datasets. This paper introduces and will focus on the new data expansion from 85 to 128 datasets. Beyond expanding this valuable resource, this paper offers pragmatic advice to anyone who may wish to evaluate a new algorithm on the archive. Finally, this paper makes a novel and yet actionable claim: of the hundreds of papers that show an improvement over the standard baseline (1-Nearest Neighbor classification), a large fraction may be misattributing the *reasons* for their improvement. Moreover, they may have been able to achieve the same improvement with a much simpler modification, requiring just a single line of code.

Keywords: Time Series · Data Mining · UCR Time Series Archive · Benchmarking

1 Introduction

The discipline of time series data mining dates back to at least the early 1990s (Agrawal, Faloutsos, and Swami 1993). As noted in a survey (Keogh and Kasetty 2003), during the first decade of research, the vast majority of papers tested only on a single artificial

dataset created by the proposing authors themselves (Agrawal, Faloutsos, and Swami 1993; Huang and Yu 1999; Kim, Lam, and Han 2000; Saito and Coifman 1995). While this is forgivable given the difficulty of obtaining data in the early days of the web, it made gauging progress and the comparisons of rival approaches essentially impossible. Frustrated by this difficulty (Keogh and Kasetty 2003), and inspired by the positive contributions of the more general UCI Archive to the machine learning community (Lichman 2013), Keogh & Folias introduced the UCR Archive in 2002 (Keogh and Folias 2002). The last expansion took place in 2015, bringing the number of the datasets in the archive to 85 datasets (Chen et al. 2015). As of Fall 2018, the archive has about 850 citations, but perhaps twice that number of papers use some fractions of the dataset unacknowledged[1].

While the archive is heavily used, it has invited criticisms, both in published papers (Hu, Chen, and Keogh 2016), and in informal communications to the lead archivists (i.e. the current authors). Some of these criticisms are clearly warranted, and the 2018 expansion of the archive (Dau, Keogh, et al. 2018) that accompanies this paper is designed to address some of the issues pointed out by the community. In addition, we feel that some of the criticisms are unwarranted, or at least explainable. We take advantage of this opportunity to, for the first time, explain some of the rationale and design choices made in producing the original archive.

The rest of this paper is organized as follows. In Section 2 we explain how the baseline accuracies that accompanies the archive are set. Section 3 enumerates the major criticisms of the archive and discusses our defense or how we have addressed the criticism with this expansion. In Section 4, we demonstrate how bad the practice of "cherry-picking" can be, allowing very poor ideas appear promising. In Section 5 we outline our best suggested practices for using the archive to produce forceful

[1] Why would someone use the archive and not acknowledge it? *Carelessness* probably explains the majority of such omissions. In addition, for several years (approximately 2006 to 2011), access to the archive was conditional on informally pledging to test on *all* datasets to avoid cherry picking (see Section 4). Some authors who did then go on to test on only a limited subset, possibly choosing not to cite the archive to avoid bringing attention to their failure to live up to their implied pledge.

classification experiments. Section 6 introduces the new archive expansion. Finally, in Section 7 we summarize our contributions and provide directions for future work.

2 Setting the Baseline Accuracy

From the first iteration, the UCR Archive has had a single predefined train/test split, and three baseline ("strawman") scores accompany it. The baseline accuracies are from the classification result of the 1-Nearest Neighbor classifier (1-NN). Each test exemplar is assigned the class label of its closest match in the training set. The notion of "closest match" is how similar the time series are under some distance measures. This is straightforward for Euclidean distance (ED), in which the data points of two time series are linearly mapped i^{th} value to i^{th} value. However, in the case of the Dynamic Time Warping distance (DTW), the distance can be different for each setting of the warping window width, known as the warping constraint parameter w (Dau, Silva, et al. 2018).

DTW allows non-linear mapping between time series data points. The parameter w controls the maximum lead/lag for which points can be mapped to, thus preventing pathological mapping between two time series. The data points of two time series can be mapped i^{th} value to j^{th} value, with $|i - j| \leq s$, where s is some integers, typically a small fraction of the time series length. In practice, this parameter is usually expressed as a percentage of the time series length and therefore, having values between 0 - 100%. The use of DTW with w = 100% is called DTW with no warping window, or unconstrained DTW. The special case of DTW with w = 0% degenerates to the ED distance.

The setting of w can have a significant effect to the clustering and classification result (Dau, Silva, et al. 2018). If not set carefully, a poor choice for this parameter can drastically deteriorate the classification accuracy. For most problems, a w greater than 20% is not needed and likely only imposes a computational burden.

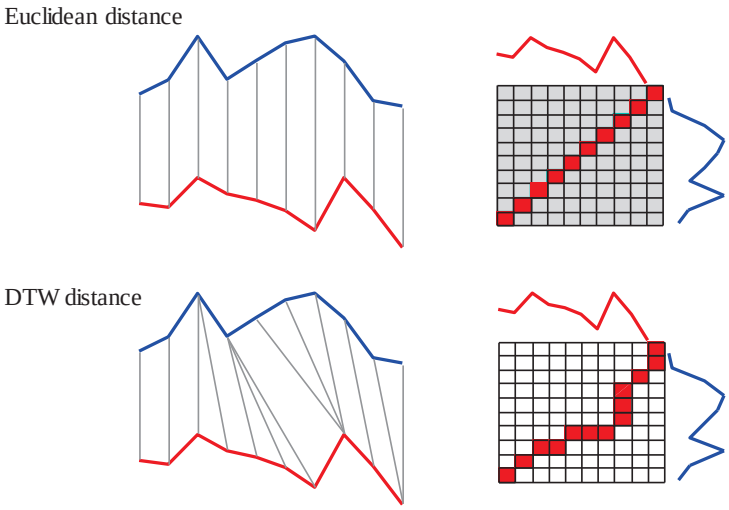

Fig. 1 Visualization of the warping path. *top*) Euclidean distance with one-to-one point matching. The warping path is strictly diagonal (cannot visit the grayed-out cells). *bottom*) unconstrained DTW with one-to-many point matching. The warping path can monotonically advance through any cell of the distance matrix.

We refer to the practice of using 1-NN with Euclidean distance as 1-NN ED, and the practice of using 1-NN with DTW distance as 1-NN DTW. The UCR Time Series Archive reports three baseline classification results. These are classification error rate of:

- 1-NN Euclidean distance
- 1-NN unconstrained DTW
- 1-NN constrained DTW with learned warping window width

For the last case, we must *learn* a parameter from the training data. The best warping window width is decided by performing Leave-One-Out Cross-Validation (LOO CV) with the train set, choosing the smallest value of *w* that minimizes the average train error rate. Generally, this approach works well in practice. However, it can produce poor results as in some situations, the best *w* in training may not be the best *w* for testing. The *top* row of Fig. 2 shows some examples where the learned constraint closely predicts the effect the warping window will have on the unseen data. The *bottom* row of Fig. 2, in contrast, shows some examples where the learned constraint fails to track the real test error rate, thus giving non-optimal classification result on holdout data.

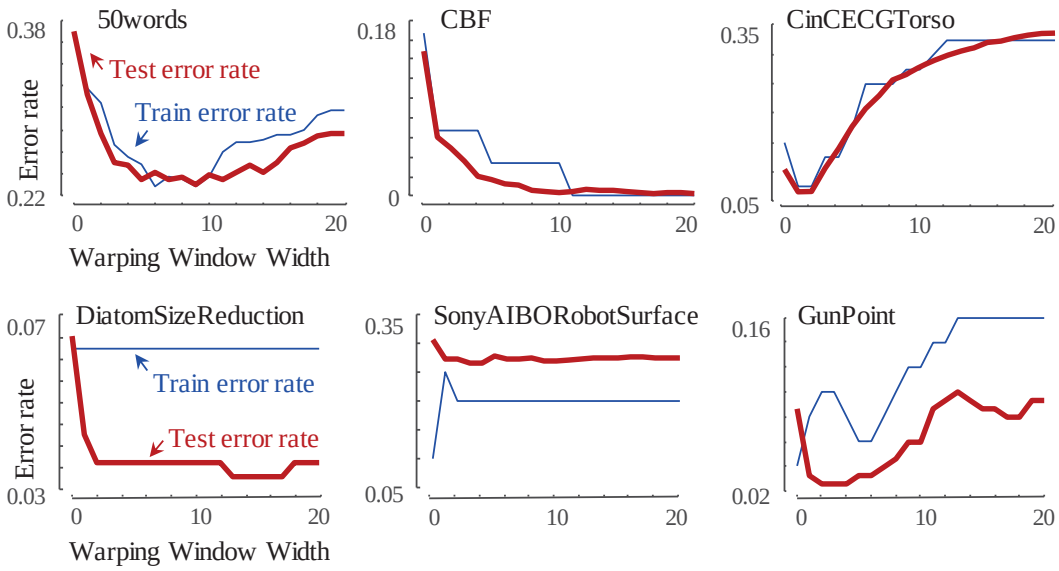

Fig. 2 blue/fine) The leave-one-out error rate for increasing values of warping window *w*, using DTW-based 1-nearest neighbor classifier. red/bold) The holdout error rate. In the bottom-row examples, the holdout accuracies do not track the predicted accuracies.

Happily, the former case is much more common (Dau, Silva, et al. 2018). When does learning the parameter fail? Empirically, the problem only occurs for very small training sets, however this issue is common in real world deployments.

3 Criticism of the UCR Archive

In this section we consider the criticisms that have been levelled at the UCR Archive. We enumerate and discuss them in no particular order.

3.1 Unrealistic assumptions

Bing et al. have criticized the archive for the following unrealistic assumptions (Hu, Chen, and Keogh 2016).

- *There is a copious amount of perfectly aligned atomic patterns.* However, in at least in some domains, labeled training data can be expensive or difficult to obtain.
- *The patterns are all of equal length.* In practice, many patterns reflecting the same behavior can be manifest at different lengths. For example, a natural walking gait cycle can vary by at least plus or minus 10%.

- *Every item in the archive belongs to exactly one well-defined class; there is no option to choose an "`unknown`" or "`unclassifiable`".* For example, in the *Cricket* datasets, each signal belongs to one of twelve classes, representing the hand signs made by an umpire. However, for perhaps 99% of a game, the umpire is not making *any* signal. It can be argued that any practical system needs to have a thirteenth class named "`not-a-sign`". This is not trivial, as *this* class will be highly variable, and this would create a highly skewed dataset.

3.2 The provenance of the data is poor

Here we can only respond *mea culpa*. The archive was first created as a small-scale personal project for Keogh's lab at University of California, Riverside. We did not know at the time that it would expand so large and become an important resource for the community. In this release, we attempt to document the datasets in a more systematic manner. In fact, one of the criteria for including a new dataset in the archive is that it has a detailed description from the data donor or it has been published in a research paper that we could cite.

3.3 Data already normalized

The time series are already z-normalized to remove offset and scaling (transformed data have zero mean and in unit of standard deviation). The rationale for this step was previously discussed in the literature (Rakthanmanon et al. 2013); we will briefly review it here with an intuitive example.

Consider the *GunPoint* dataset shown in Fig. 6. Suppose that we did not z-normalize the data but allowed our classifier to exploit information about the exact *absolute* height of the gun or hand. As it happens, this *would* help a little. However, imagine we collected more test data next week. Further suppose that for this second session, the camera zoomed in or out, or the actors stood a little closer to the camera, or that the female actor decided to wear new shoes with a high heel. None of these differences would affect z-normalized data as z-normalization accounts for offset and scale variance; however, they would drastically (negatively) affect any algorithm that exploited the raw unnormalized values.

Nevertheless, we acknowledge that for some (we believe, *very* rare) cases, data normalization is ill-advised. For the new datasets in this release, we provide the raw data without any normalization when possible; we explicitly state if the data has been normalized beforehand by the donors (the data might have been previously normalized by the donating source, who lost access to original raw data).

3.4 The individual datasets are too small

The largest dataset is *StarLightCurves* with 1,000 train and 8,236 test objects, covering 3 classes. The smallest dataset is *Beef* with 30 train and 30 test objects, covering 5 different classes. Note that in recent years, there have been several published papers that say something to the effect of "*in the interests of time, we only tested on a subset of the UCR Archive*". While it is true that there is a need for bigger datasets in the era of "big data" (some algorithms specifically target scaling for big datasets), the archive has catered a wide array of data mining needs and lived up to its intended scope. Perhaps a specialist archive of massive time series can be made available for the community in a different repository.

3.5 The datasets are not reflective of real-world problems

This criticism is somewhat warranted. The archive is biased towards:

- Datasets that reflect the personal interests/hobbies of the principal investigator (PI), Eamonn Keogh, including entomology (*InsectWingbeatSound*), anthropology (*ArrowHead*) and astronomy (*StarLightCurves*). A wave of datasets added in 2015 reflect the personal and research interests of Tony Bagnall (Bagnall et al. 2018), many of which are image-to-time-series datasets. The archive has always had a policy of adding *any* donated dataset, but offers of donations are surprisingly rare. Even when we actively solicited donations by writing to authors and asking for their data, we found that only a small subset of authors is willing to share data. The good news is that there appears to be an increasing willingness to share data, perhaps thanks to conferences and journals actively encouraging reproducible research.
- Datasets that could be easily obtained or created. For example, fMRI data could be very interesting to study, but the PI did not have access to such a machine or

the domain knowledge to create a classification dataset in this domain. However, with an inexpensive scanner or a camera, it was possible to create many image-derived datasets such as *GunPoint*, *OSULeaf*, *SwedishLeaf*, *Yoga*, *Fish* or *FacesUCR*.

- Datasets that do not have privacy issues. For many data types, mining the data while respecting privacy is an important issue. Unfortunately, none of the datasets in the UCR Archive motivates the need for privacy (though it is possibly to use the data to construct proxy datasets).

3.6 Benchmark results are from a single train/test split

Many researchers, especially those coming from a traditional machine learning background have criticized the archive for having a single train/test split. The original motivation for fixing the train and test set was to allow *exact* reproducibility. Suppose we simply suggested doing five-fold cross validation. Further suppose, someone claimed to be able to achieve an accuracy of *A*, on some datasets in the archive. If someone else reimplemented their algorithm and got an accuracy that is slightly lower than *A* during their five-fold cross validation, it would be difficult to know if that was within the expected variance of different folds, or the result of a bug or a misunderstanding in the new implementation. This issue would be less of a problem if everyone shared their code, and/or had very explicit algorithm descriptions. However, while the culture of open source code is growing in the community, such openness was not always the norm.

With a single train/test split, and a deterministic algorithm such as 1-NN, failure to *exactly* reproduce someone else's result immediately suggests an issue that should be investigated before proceeding with research. Note that performing experiments on the single train/test split was always suggested as an absolute minimum sanity check; it did not/does not preclude pooling the two splits and then performing *K*-fold cross validation or any other more rigorous evaluation.

4 How Bad is Cherry Picking?

It is not uncommon to see papers which report only results on a subset of the UCR Archive, without any justification or explanation. Here are some examples.

- "*We evaluated 1D-SAXLSSS classification accuracy on 22 datasets (see Table 2) taken from publicly available UCR repository benchmark*" (Taktak, Triki, and Kamoun 2017)
- "*Figure 3 shows a performance gain of DSP-Class-SVM and DSP-Class-C5.0 approach in 5/11 datasets compared to another technique that does not use features (1NN with Euclidean distance)*" (A. Silva and Ishii 2016).
- "*We experiment 48 small-scale datasets out of total 85 problems in the UCR time series archive*" (He et al. 2018)

We will show how cherry picking can make a vacuous idea look good. However, to be clear we are not suggesting that the works considered above are in any way disingenuous.

Consider the following section of text (italicized for clarity) with its accompanying table and figure, and imagine it appears in published report. While this is a fictional report, note that all the numbers presented in the table and figure are *true* values, based on reproducible experiments that we performed.

We tested our novel FQT algorithm on 20 datasets from the UCR archive. We compared to the Euclidean Distance, a standard benchmark in this domain. Table T summarizes the results numerically, and Figure F shows a scatterplot visualization.

Table T: Performance comparison between Euclidean distance and our FQT distance. Our proposed FQT distance wins on all datasets that we consider.

Data Set	ED Error	FQT Error	Error Reduction
Strawberry	0.062	0.054	0.008
ECG200	0.120	0.110	0.010
TwoLeadECG	0.253	0.241	0.012
Adiac	0.389	0.376	0.013
ProximalPhalanxTW	0.292	0.278	0.014
DistalPhalanxTW	0.273	0.258	0.015
ProximalPhalanxOutlineCorrect	0.192	0.175	0.017
RefrigerationDevices	0.605	0.587	0.018
Wine	0.389	0.370	0.019
ProximalPhalanxOutlineAgeGroup	0.215	0.195	0.020
Earthquakes	0.326	0.301	0.025
ECGFiveDays	0.203	0.177	0.026
SonyAIBORobotSurfaceII	0.141	0.115	0.026
Lightning7	0.425	0.397	0.028
Trace	0.240	0.210	0.030
MiddlePhalanxTW	0.439	0.404	0.035
ChlorineConcentration	0.350	0.311	0.039

BirdChicken	0.450	0.400	0.050
Herring	0.484	0.422	0.062
CBF	0.148	0.080	0.068

Note that we used identical (UCR pre-defined) splits for both approaches, and an identical classification algorithm. Thus, all improvements can be attributed to our novel distance measure. The improvements are sometimes small, however, for CBF, Herring and BirdChicken they are 5% (0.05) or greater, demonstrating that our FQT distance measure potentially offers significant gains in some domains. Moreover, we prove that FQT is a metric, and therefore easy to index with standard Tree-Access-Methods.

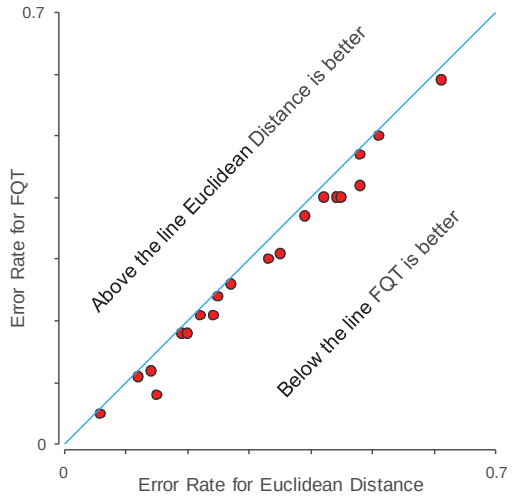

Figure F: The error rate of our FQT method compared to Euclidean distance. Our proposed method clearly outperforms the baseline for all datasets that we consider.

(Returning to the current authors voice)

The above results are all true, and the authors are correct in saying that FQT is a metric and is easier to index than DTW. So, what is this remarkable FQT distance measure? It is simply the Euclidean distance after the first 25% of each time series is thrown away (First Quarter Truncation). Here is how we compute the FQT distance for time series A and B in MATLAB:

```
FQT_dist = sqrt(sum((A(end*0.25:end) - B(end*0.25:end)).^2))
```

If we examine the full 85 datasets in the archive, we will find that FQT wins on 19 datasets, but loses/draws on 66 (if we count a win as at least 1% reduction in error rate).

Moreover, the size of the losses is generally more dramatic than the wins. For instance, the "error reduction" for `MedicalImages` is -0.23 (accuracy decreases by 23%).

Simply deleting the first quarter of every time series is obviously not a clever thing to do, and evaluating this idea on all the datasets confirms that. However, by cherry picking the twenty datasets that we chose to report, we made it seem like *very* good idea. It is true that in a full paper based on FQT we would have had to explain the measure, and it would have struck a reader as simple, unprincipled and unlikely to be truly useful. However, there are a vast number of algorithms that exist that would have the same basic "*a lot worse on most, a little better on a few*" outcome, and many of these could be framed to sound like plausible contributions (cf. Section 5.1).

In a recent paper, Lipton & Steinhardt list some *Troubling Trends in Machine Learning Scholarship* (Lipton and Steinhardt 2018). One issue identified is "mathiness", defined as "*the use of mathematics that obfuscates or impresses rather than clarifies*". We have little doubt that we could "dress up" our proposed FQT algorithm with spurious notation ((Lipton and Steinhardt 2018) call it "*fancy mathematics*") to make it sound complex.

To summarize this section, cherry picking can make an arbitrary poor idea look useful, or even wonderful. Clearly, not all (or possibly even, not *any*) papers that report on a subset of the UCR datasets are trying to deceive the reader. However, as an outsider to the research effort, it is essentially impossible to know if the subset selection was random and fair (made *before* any results were computed) or biased to make the approach appear better than it really is.

The reasons given for testing only on a subset of the data (where any reason is given at all) is typically something like "*due to space limitations, we report only five of the…*". However, this is not justified. A Critical Different Diagram like Fig. 4 or a scatter plot like Fig. 5 requires very little space but can summarize an arbitrary number of datasets. Moreover, one can always place detailed spreadsheets online or in an accompanying, cited technical report, as many papers do these days (Paparrizos and Gravano 2015; Dau, Silva, et al. 2018).

That being said, we believe that sometimes there are good reasons to test a new algorithm on only a subset of the archive, and we applause researchers who explicitly justify their conduct. For example, Hills et al. stated: "*We perform experiments on 17*

datasets from the UCR time-series repository. We selected these particular UCR datasets because they have relatively few cases; even with optimization, the shapelet algorithm is time consuming." (Hills et al. 2014).

5 Best Practices for Using the Archive

Beating the performance of DTW on some datasets should be considered a *necessary*, but not *sufficient* condition for introducing a new distance measure or classification algorithm. This is because the performance of DTW itself can be improved with very little effort, in at least a dozen ways. In many cases, these simple improvements can close most or all the gap between DTW and the more complex measures being proposed. For example:

- The warping window width parameter of cDTW algorithm is tuned by the "quick and dirty" method described in Section 2. As Fig. 2.*bottom.row* shows, on at least some datasets, that tuning is suboptimal. The parameter could be tuned more carefully in several ways such as by resampling or by creating synthetic examples (Dau et al. 2017).
- The performance of DTW classification can often be improved by other trivial changes. For example, as shown in Fig. 3.*left*, simply smoothing the data can produce significant improvements. Fig. 3.*right* shows that generalizing from one-nearest neighbor to k-nearest-neighbor often helps. One can also test alternative DTW step patterns (Lu et al. 2017). Making DTW "endpoint invariant" helps significantly on many datasets (D. F. Silva, Batista, and Keogh 2017), etc.

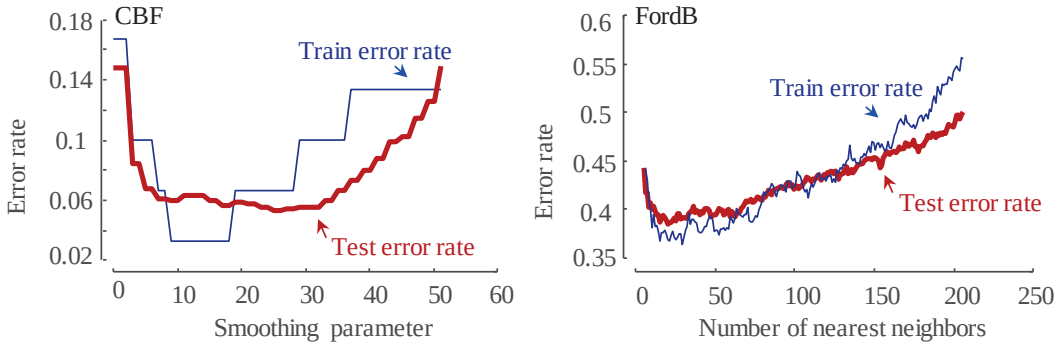

Fig. 3 left) The error rate on classification on the *CBF* dataset for increasing amounts of smoothing using MATLAB's default smoothing algorithm. right) The error rate on classification on the *FordB* dataset for increasing number of nearest neighbors. Note that the leave-one-out error rate on the training data does approximately predict the best parameter to use.

An hour spent on optimizing any of the above will improve the performance of ED/DTW on a significant fraction of the archive.

5.1 Misattribution of improvements: a cautionary tale

We believe that of the several hundred papers that show an improvement on the baselines for the UCR Archive, a significant fraction is misattributing the cause of their improvement and is simply *indirectly* discovering one of the low hanging fruits above. This point has recently been made in the more general case by researchers who believe that many papers suffer from a failure "*to identify the sources of empirical gains*" (Lipton and Steinhardt 2018). Below we show an example to demonstrate this.

Many papers have suggested using a wavelet representation for time series classification[2], and have gone on to show accuracy improvements over either the DTW or ED baselines. In most cases, these authors attribute the improvements to the multi-resolution properties of wavelets. For example (our emphasis in the quotes below):

- "*wavelet compression techniques can sometimes even help achieve higher classification accuracy than the raw time series data, as they better capture essential local features… As a result, we think it is safe to claim that multi-level*

[2] These works should not be confused with papers that suggest using a wavelet representation to perform dimensionality reduction to allow more efficient indexing of time series.

wavelet transformation is indeed helpful for time series classification." (Li, Bissyande, Klein, and Le Traon 2016)

- "*our multi-resolution approach as discrete wavelet transforms have the ability of reflecting the local and global information content at every resolution level.*" (Li, Bissyande, Klein, and Le Traon 2016)
- "*We attack above two problems by exploiting the multi-scale property of wavelet decomposition … extracting features combining the global information and partial information of time series.*" (Zhang et al. 2006)
- "*Thus multi-scale analyses give us the ability of observing time series in various views.*" (Zhang and Ho 2005)

As the quotes above suggest, many authors attribute their accuracy improvements to the multi-resolution nature of wavelets. However, we have a different hypothesis. The wavelet representation is simply smoothing the data implicitly, and all the improvements can be attributed to *just* this smoothing! Fig. 3 above does offer evidence that at least on some datasets, appropriate smoothing is enough to make a difference that is commeasure with the claimed improvements. However, is there a way in which we could be sure? Yes, we can exploit an interesting property of the Haar DWT.

Note that if both the original and reduced dimensionality of the Haar Wavelet transform are integer powers of two, then the approximation produced by Haar is logically *identical* to the approximation produced by the Piecewise Aggerate Approximation (Keogh et al. 2001). This means that the distances calculated in the truncated coefficient space are identical for both approaches, and thus they will have the same classification predictions *and* the same error rate. If we revisit the *CBF* dataset shown in Fig. 3, using Haar with 32 coefficients we get an error rate of just 0.05, much better than the 0.148 we would have obtained using the raw data. Critically, however, PAA with 32 coefficients also gets the same 0.05 error rate.

It is important to note that PAA is not in any sense *multi-resolution* or *multiscale*. Moreover, by the definition of PAA, each coefficient being the *average* of a *range* of points, is very similar to the definition of the moving average filter smoothing, with each point being *averaged* with its neighbors within a *range* to the left and the right.

We can do one more test to see if the Haar Wavelet is offering us something beyond smoothing. Suppose we use it to classify data that has *already* been smoothed with a smoothing parameter of 32. Recall (Fig. 3) that using this smoothed data directly gives us an error rate of 0.055. Will Haar Wavelet classification further improve this result? No, in fact using 32 coefficients, both Haar Wavelet and PAA classification produce very slightly worse results of 0.057, presumably because we have effectively smoothed the data twice, and by doing so, oversmoothed it (again, see Fig. 3, and examine the trend of the curve as the smoother parameter grows above 30). In our view, these observations cast significant doubt on the claim that the improvements obtained can correctly be attributed to multi-resolution properties of wavelets.

There are two obvious reasons as two why this matters.

- Misattribution of *why* a new approach works potentially leaves adopters in the position of fruitless follow-up work or application of the proposed ideas to data/tasks for which they are not suited.
- If the problem at hand is really to improve the accuracy of time series classification, and if five minutes spent experimenting with a smoothing function can give you the same improvement as months implementing a more complex method, then surely the former is more desirable, if less publishable.

This is simply one concrete example. We suspect that there are many other examples. For example, many papers have attributed time series classification success to their exploiting the "memory" of Hidden Markov Models, or "long-term dependency features" of Convolution Neural Networks etc. However, one almost never sees an ablation study that forcefully convinces the reader that the *claimed* reason for improvement is correct.

In fairness, a handful of papers do explicitly acknowledge that. While they may introduce a complex representation or distance measure for classification, at least some of the improvements should be attributed to smoothing. For example, Schäfer notes: "*Our Bag-of-SFA-Symbols (BOSS) model combines the extraction of substructures with the tolerance to extraneous and erroneous data using a noise reducing representation of the time series*" (Schäfer 2015). Likewise, Li and colleagues (Li, Bissyande, Klein, and Traon 2016) revisit their Discrete Wavelet Transformed (DWT) time series

classification work (Li, Bissyande, Klein, and Le Traon 2016) to explicitly ask "*if the good performances of DWT on time series data is due to the implicit smoothing effect*". They show that their previous embrace of Wavelet-based classification does not produce results that are better than simple smoothing in a statistically significant way. Such papers, however, remain an exception.

5.2 How to compare classifiers

5.2.1 The choice of metric

Suppose you have an archive of one hundred data sets and you want to test whether classifier A is better than classifier B, or compare a set of classifiers, over these data. Firstly, you need to specify what you mean by one classifier being "better" than another. There are two general criteria with which we may wish compare classifier ability on data not used in the training process: the prediction ability and the probability estimates.

The ability to predict is most commonly measured by accuracy (or equivalently, error rate). However, accuracy does not always tell the whole story. If a problem has class imbalance, then accuracy may be less informative than a measure that compensates for this skewed classes. Sensitivity, specificity, precision, recall and the F statistic are all commonly used for two class problems where one class is rarer than the other, such as medical diagnosis trials (Okeh and Okoro 2012; Uguroglu 2013). However, these measures do not generalize well to multiclass problems or scenarios where we cannot prioritize one class using domain knowledge.

For the general case over multiple diverse datasets, we consider accuracy and balanced accuracy enough to assess predictive power. Conventional accuracy is the proportion of examples correctly classified while balanced accuracy is the average of accuracy for each class individually (Brodersen et al. 2010). Some classifiers produce probability estimates for each class and these can be summarized with statistics such as negative log-likelihood or area under the receiver operator curve. However, if we are using a classifier that only produces predictions (such as 1-NN), these metrics are inappropriate.

5.2.2 The choice of data split

Having decided on a comparison metric, the next issue is what data you are going to evaluate the data on. If a train/test split is provided, it is natural to start by building all the classifiers (including all model selection/parameter setting) on the train data, and then assess accuracy on the test data. There are two main problems with using a single train test split to evaluate classifiers.

First, there is a temptation to cheat by setting the parameters to optimize the test data. This can be explicit, for example, by setting an algorithm to stop learning when test accuracy is maximized, or implicit, by setting default values based on knowledge of the test split. For example, suppose I have generated results such as Fig. 3, and have a variant of DTW I wish to assess on this test data. I may perform some smoothing and set the parameter to a default of 10. Explicit bias can only really be overcome with complete code transparency. For this reason, we *strongly* encourage users of the archive to make their code available to both reviewers and readers.

The other problem with a single train/test split, particularly with small datasets, is that tiny differences in performance can seem magnified. For example, we were recently contacted by a researcher who queried our published results for one Nearest Neighbor (1-NN) Dynamic Time Warping (DTW) on the UCR repository train/test splits. When comparing our accuracy results to theirs, they noticed that in some instances they differ by as much as 6%. When we compared the results for the other problems there was no significant difference, but clearly, we were concerned by this single difference, as the algorithm in question is deterministic. On further investigation, we found out that our data were rounded to six decimal places, theirs to eight. These differences on single splits were caused by small data set sizes and tiny numerical differences (often *just* a single case classified differently).

These problems can be largely overcome by merging the train and test data and resampling each dataset multiple times then averaging test accuracy. If this is done, there are several caveats:

- the default train and test splits should always be included as the first resample;
- resamples must be the same for each algorithm;
- the resamples should retain the initial train and test sizes;

- and the resamples should be stratified to retain the same class distribution as the original.

Even when meeting all these constraints, resampling is not always appropriate. Some data are constructed to keep experimental units of observation in difference datasets. For example, when constructing the alcohol fraud detection problem (Lines, Taylor, and Bagnall 2016), we used different bottles in the experiments and made sure that observations from the same bottle does not appear in both the train and the test data. We do this to make sure we are not detecting bottle differences rather than different alcohol levels. Problems such as these discourage practitioners from resampling. However, we note that the majority of machine learning research involves repeated resamples of the data.

Finally, for clarity we will repeat our explanation in Section 3.6 as to why we have a single train/test split in the archive. Publishing the results of both ED and DTW distance on a single deterministic split allows researchers a useful sanity check, before they perform more sophisticated analysis (Salzberg 1997; Demšar 2006; Garcia and Herrera 2008).

5.2.3 The choice of significance tests

Whether through a single train/test split or through resampling then averaging, you now arrive at a position of having multiple accuracy estimates for each classifier. The core question is, are there significant differences between the classifiers? In the simpler scenario, suppose we have two classifiers, and we want to test whether the differences in average accuracy is different to zero. There are two alternative hypothesis tests that one could go for, a paired two-sample t-test (Student 1908) to test whether there is evidence of a significant difference in mean accuracies or a Wilcoxon signed-rank test (Wilcoxon 1945; Siegal 1956) for differences in median accuracies. Generally, machine learning researchers favor the latter. However, it is worth noting that many of the problems identified with parametric tests in machine learning derive from the problem of two few datasets, typically twenty or less. With more than 30 datasets, the central limit theorem means these problems are minimized. Nevertheless, we advise using a Wilcoxon signed-rank test with a significance level of α set at or smaller than 0.05 to satisfy reviewers.

What if you want to compare multiple classifiers on these datasets? We follow the recommendation of Demšar (Demšar 2006) and base the comparison on the ranks of the classifiers on each dataset rather than the actual accuracy values. We use the Friedmann test (Friedman 1937, 1940) to determine if there were any statistically significant differences in the rankings of the classifiers. If differences exist, then the next task is to determine where they lie. This is done by forming cliques, which are groups of classifiers within which manifest significant difference. Following recent recommendations in (Benavoli, Corani, and Mangili 2016) and (Garcia and Herrera 2008), we have abandoned the Nemenyi post-hoc test (Hollander and Wolfe 1999) originally used by Demšar (Demšar 2006). Instead, we compare all classifiers with pairwise Wilcoxon signed-rank tests and form cliques using the Holm correction, which adjusts family-wise error less conservatively than a Bonferonni adjustment (Holm 1979). We can summarize these comparisons in a critical difference diagram such as Fig. 4.

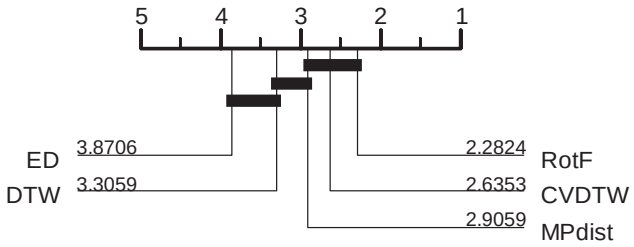

Fig. 4 Critical difference for MPdist distance against four benchmark distances. Figure credited to (Gharghabi et al. 2018). We can summarize this diagram as follow: RotF is the best performing algorithm with an average rank of 2.2824; there is an overall significant difference among the five algorithms; there are three distinct cliques; MPdist is significantly better than ED distance and not significantly worse than the rest.

Fig. 4 displays the performance comparison between MPdist, a new distance measure and other competitors (Gharghabi et al. 2018). This diagram orders the algorithms and presents the average rank on the number line. Cliques of methods are grouped with a solid bar showing groups of methods within which there is no significant difference. According to Fig. 4, the best ranked method is Rotation Forest (RotF), however, it is not statistically better than the other two methods in its clique, CVDTW and MPdist.

5.3 A checklist

We propose the following checklist for any researcher who is proposing a novel time series classification/clustering technique and would like to test it on the UCR Archive.

1. Did you test on *all* the datasets? If not, you should carefully explain why not, to avoid the appearance of cherry picking. For example, "*we did not test on the large datasets, because our method is slow*" or "*our method is only designed for ECG data, so we only test on the relevant datasets*".
2. Did you take care to optimize all parameters on *just* the training set? Producing a Texas Sharpshooter plot is a good way to visually confirm this for the reviewers (Batista et al. 2014).
3. Do you perform an appropriate statistical significance test? (see Section 5.2.3)
4. If you are claiming your approach is better due to property X, did you conduct an ablation test (lesion study) to show that if you remove this property, the results worsen, and/or, if you endow an otherwise unrelated approach with this property, that approach also improves?
5. Did you share your code? Note, some authors state in their submission, something to the effect of "*We will share code when the paper gets accepted*". However, sharing of carefully documented code at the time of submission is the best way to imbue confidence in even the most cynical reviewer.
6. If you modified the data in any way (adding noise, smoothing, interpolating, etc.), did you share the modified data, or the code, with random seed generator, that would allow a reader to exactly reproduce the data.

6 The New Archive

On January 2018, we reached out about forty researchers soliciting ideas for the new UCR Time Series Archive. They are among the most active researchers in the time series data mining community, who have used the UCR Archive in the past. We raised the question: "*What would you like to see in the new archive*?". We saw a strong consensus on the following needs:

- Longer time series
- Variable length datasets

- Multi-variate datasets
- Information about the provenance of the datasets

Some researchers also wish to see the archive to include datasets suitable for some specific research problems. For example:

- Datasets with highly unbalanced classes
- Datasets with very small training set to benchmark data augmentation techniques

Researchers especially raised the need for bigger datasets in the era of big data.

> "*To me, the thing we lack the most is larger datasets; the largest training data has 8,000 time series while the community should probably move towards millions of examples. This is a wish, but this of course doesn't go without problems: how to host large datasets, will future researchers have to spend even more time running their algorithms, etc*." (Francois Petitjean, Monash University)

Some researchers propose sharing pointers to genuine data repositories and data mining competitions.

> "*A different idea that might be useful is to add dataset directories that have pointers to other datasets that are commonly used and freely available. When the UCR Archive first appeared, it was a different time, with fewer high quality, freely available datasets that were used by many researchers to compare results. Today there are many such datasets, but you tend to find them with Google, or seeing mentions in papers. One idea would be to pull together links to those datasets in one location with a flexible "show me datasets with these properties or like this dataset" function*." (Tim Oates, University of Maryland Baltimore County).

While some of these ideas may go beyond the ambition of the UCR Archive, we think that it inspires a wave of effort in making the time series community better. We hope *others* will follow suit our effort in making datasets available for research purposes. In a sense, we think it would be inappropriate for our group to provide all such needs, as this monopoly might bias the direction of research to reflect our interests and skills.

We have addressed *some* of the perceived problems with the existing archive and some of what the community want to see in the new archive. We follow with an introduction of the new archive release.

6.1 General introduction

We refer to archive before the Fall 2018 expansion as the *old* archive and the current version as the *new* archive. The Fall 2018 expansion increases the number of datasets from 85 to 128. We adopt a standard naming convention, that is using captions for words and no underscores. Where possible, we include the provenance of the datasets. In editing data for the new archive, in most cases, we make the test set bigger than the train set to reflect real-world scenarios, i.e., labeled train data are usually expensive. We keep the datasets as is if they are from a published papers and are already in a UCR Archive preferred format (see guidelines for donating datasets to the archive (Dau, Keogh, et al. 2018)).

In Fig. 5. we use the Texas Sharpshooter plot popularized by Batista et al. (Batista et al. 2014) to show the baseline result comparison between 1-NN ED and 1-NN constrained DTW on 128 datasets of the new archive.

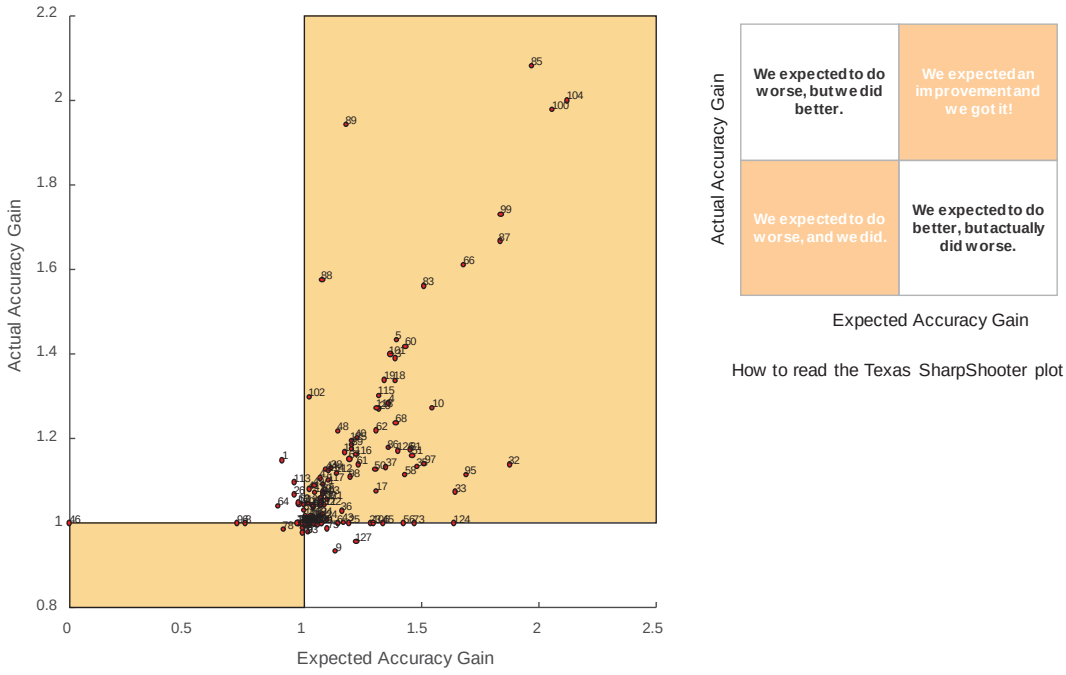

Fig. 5 Comparison of Euclidean distance versus constrained DTW for 128 datasets. In the Texas Sharpshooter plot, each dataset falls into four possibilities (see the interpretation on the right). We optimize the performance of DTW by learning a suitable warping window width and compare the expected improvement with the actual improvement. The results are strongly supportive of the claim that DTW is better than Euclidean distance for most problems. Note that some of the numbers are hard to read because they overlap. A higher resolution version of this image is available at the UCR Archive webpage (Dau, Keogh, et al. 2018).

The Texas Sharpshooter plot is introduced to avoid the Texas sharpshooter fallacy (Batista et al. 2014), that is a simple logic error that seems pervasive in time series classification papers. Many papers show that their algorithm/distance measure are better than the baselines/competitors on some datasets, ties on many and loses on some. They then claim their method works for some domains and thus it has value. However, it is not useful to have an algorithm that are good for some problems unless you can tell *in advance* which problems they are. The Texas Sharpshooter plot in Fig. 5 compares ED and constrained DTW distance by showing the expected accuracy gain (based solely on train data) versus the actual accuracy gain (based solely on test data) of the two methods. Note that here, the improvement of constrained DTW over ED is almost tautological, as constrained DTW subsumes ED as a special case. More generally, these plots visually summarize the strengths and weaknesses of rival methods.

6.2 Some notes on the old archive

We reverse the train/test split of fourteen datasets to make them consistent with their original release, i.e. when they were donated to the archive, according to the wish of the original donors. These datasets were accidentally reversed during the 2015 expansion of the archive. The train/test split of these dataset now agree with the train/test split hosted at the UEA Archive (Bagnall et al. 2018), and are the split that was used in a recent influential survey paper titled "*The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances*" (Bagnall et al. 2017). We list these datasets in Table 1.

Table 1: Fourteen datasets that have the train/test split reversed for the new archive expansion.

Dataset name	
DistalPhalanxOutlineAgeGroup	*DistalPhalanxOutlineCorrect*
DistalPhalanxTW	*Earthquakes*
FordA	*FordB*
HandOutlines	*MiddlePhalanxOutlineAgeGroup*
MiddlePhalanxOutlineAgeCorrect	*MiddlePhalanxTW*
ProximalPhalanxTW	*Strawberry*
Worms	*WormsTwoClass*

Among the 85 datasets of the old archive, there are twelve datasets that at least one algorithm gets 100% accuracy (Lines, Taylor, and Bagnall 2018; Bagnall et al. 2018). We list them in Table 2.

Table 2: Twelve "solved" datasets, which at least one algorithm gets 100% accuracy.

Type	Dataset name	Type	Dataset name
Image	*BirdChicken*	Sensor	*Plane*
Spectrograph	*Coffee*	Simulated	*ShapeletSim*
ECG	*ECGFiveDays*	Simulated	*SyntheticControl*
Image	*FaceFour*	Sensor	*Trace*
Motion	*GunPoint*	Simulated	*TwoPatterns*
Spectrograph	*Meat*	Sensor	*Wafer*

6.3 Dataset Highlights

6.3.1 GunPoint datasets

The original *GunPoint* dataset was created by current authors Ratanamahatana and Keogh in 2003. Since then, it has become the "iris" data of the time series community (Fisher 1936), being used in over one thousand papers, with images from the dataset appearing in dozens of papers (see Fig. 6). As part of these new release of the UCR archive, we decided to revisit this problem, by asking the two original actors to recreate the data.

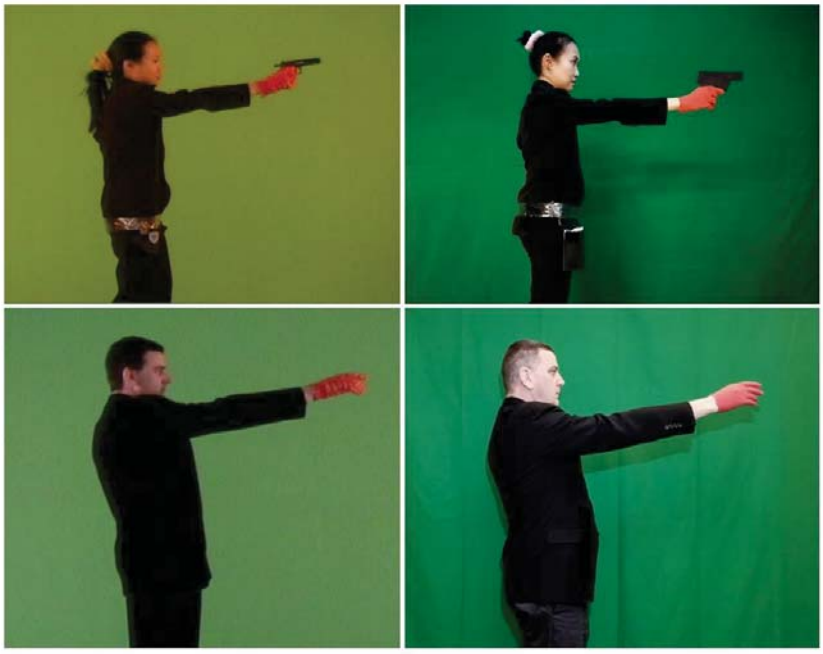

Fig. 6 *Left*) *GunPoint* recording of 2003 and *right*) *GunPoint* recording of 2018. The female and male actors are the same individuals recorded fifteen years apart.

We record two scenarios, "Gun" and "Point". In each scenario, the actors aim at a target at eye level before them. We strived to reproduce in every aspect the recoding of the original *GunPoint* dataset created 15 years ago (in 2003). The difference between Gun and Point is that in the Gun scenario, the actor holds a replica gun. They point the gun at the target, return the gun back to the waist holster and then brings their free hand to a rest position to complete an action. Each complete action conforms to a five-second cycle. We filmed with a commodity smartphone Samsung Galaxy 8. With 30fps, this

translates into 150 frames per action. We generated a time series for each action by taking the x-axis location of the centroid of the red gloved hand (see Fig. 6). We merged the data of this new recording with the old *GunPoint* data to make several new datasets. The data collected now spans two actors {F,M}, two behaviors {G,P}, and two years {03,18}. The task of the original *GunPoint* dataset was differentiating between the Gun and the Point action: {FG03, MG03} vs. {FP03, MP03}. We have created three new datasets. Each dataset has two classes; each class is highly polymorphic with four variants characterizing it.

- *GunPointAgeSpan*: {FG03, MG03, FG18, MG18} vs. {FP03, MP03, FP18, MP18}. The task is to recognize the actions with invariance to the actor, as with *GunPoint* before, but also be invariant to the year of recording.
- *GunPointOldVersusYoung*: {FG03, MG03, FP03, MP03} vs. {FG18, MG18, FP18, MP18}, which asks if a classifier can detect the difference between the recording sessions due to (perhaps) the actors aging, differences in equipment and processing; though as noted above, we tried to minimize such inconsistencies). In this case, the classifier needs to ignore the action and actor.
- *GunPointMaleVersusFemale*: {FG03, FP03, FG18, FP18} vs. {MG03, MP03, MG18, MP18}, which asks if a classifier can differentiate between the build and posture of the two actors.

6.3.2 GesturePebble datasets

The archive expansion includes several datasets whose time series exemplars can be of different lengths. The *GesturePebble* dataset is one of them. For ease of data handling, we pad enough NaNs to the end of each time series, to make it the same length of the longest time series. Some algorithms/distance measures can handle variable-length data directly; other researchers may have to process such data by truncation or renormalization etc. We deliberately refrain from offering any advice on how to best do this.

The *GesturePebble* dataset comes from the paper "*Gesture Recognition using Symbolic Aggregate Approximation and Dynamic Time Warping on Motion Data*" (Mezari and Maglogiannis 2017). This work is among the many that study the application of

commodity smart devices as a motion sensor for gesture recognition. The data is collected with the 3-axis accelerometer Pebble smart watch mounted to the participants' wrist. Each subject is instructed to perform six gestures depicted in Fig. 7. The data collection included four participants, each of which repeated the gesture set in two separate sessions a few days part. In total, there are eight recordings, which contain 304 gestures. Since the duration of each gestures varies, the time series representing each gesture are of different lengths.

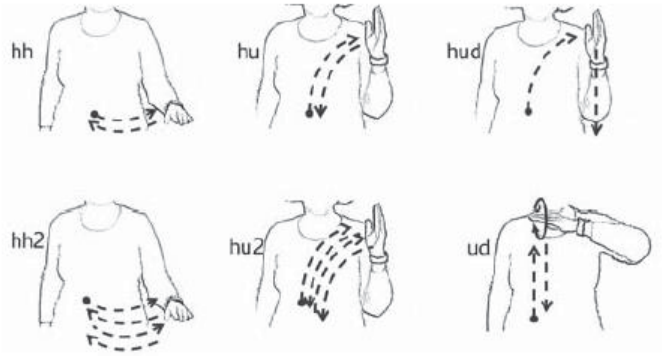

Fig. 7 The dot marks the start of a gesture. The labels (hh, hu, hud, etc) are used by original authors of the data and may not have any special meaning. The gestures are selected based on criteria that they are characterized by the wrist movements; they simple and natural enough to replicate; and they can be related to commands to control devices (Mezari and Maglogiannis 2017).

We created two datasets from the original data, both using only the z-axis reading (out of the three channels/attributes available).

- *GesturePebbleZ1*: The train set consists of data of all subjects collected in the first session. The test set consists of all data collected in the second session. This way, data of each subject appear in both train and test set.
- *GesturePebbleZ2*: The train set consists of data of two subjects and the test set consists of data of the other two subjects. This dataset is intended to be more difficult than *GesturePebbleZ1* because the subjects in the test set do not appear in the train set (presumably that each participant possesses unique gait and posture and move differently). Baseline results confirm this speculation.

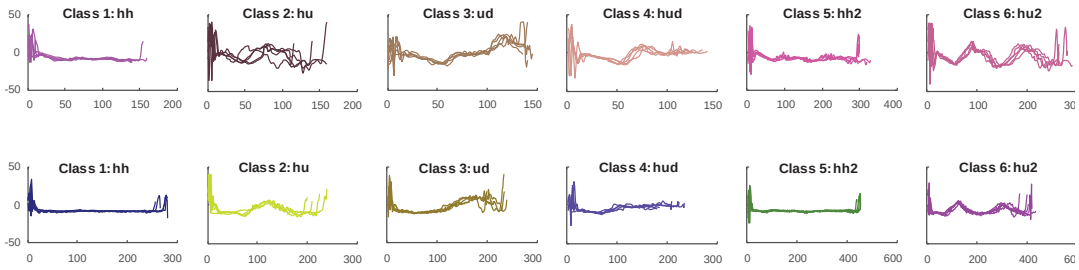

Fig. 8 Data from recording of two different subjects performing a same set of six gestures (top row and bottom row). The endpoints of the time series contain the "tap event", which are abrupt movements to signal the start and end of the gesture (deliberately performed by the subject). The accelerometer data is also under influence of gravity and the device's orientation during the movement. The end points of the time series contain tap events, which are abrupt movements the user make of their wrist to mark the start and end of a gesture.

6.3.3 EthanolLevel dataset

This dataset was produced as part of a project with Scotch Whisky Research Institute into non-intrusively detecting forged spirits (Counterfeiting whiskey an increasingly lucrative crime). One such candidate method of detecting forgeries is by examining the ethanol level extracted from a spectrograph. The dataset covers twenty different bottle types and four levels of alcohol: 35%, 38%, 40% and 45%. Each series is a spectrograph of 1,751 observations. Fig. 9 shows some examples of each class.

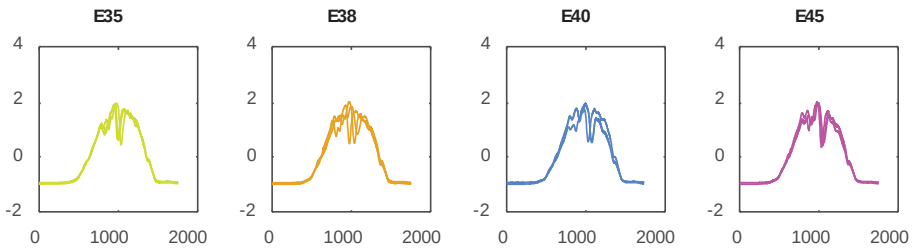

Fig. 9 Three examples per class of EthanolLevel dataset. The four classes correspond to levels of alcohol: 35%, 38%, 40% and 45%. Each series is 1751 data-point long.

This dataset is an example of when it is wrong to merge and resample, because the train/test are constructed so that the same bottle type is never in both datasets. The dataset was introduced in "*HIVE-COTE: The hierarchical vote collective of transformation-based ensembles for time series classification*" (Lines, Taylor, and Bagnall 2016).

6.3.4 *InternalBleeding datasets*

The source dataset is data from fifty-two pigs having three vital signs monitored, before and after an induced injury (Guillame-Bert and Dubrawski 2017). We created three datasets out of this source: *AirwayPressure* (airway pressure measurements), *ArtPressure* (arterial blood pressure measurements) and *CVP* (central venous pressure measurements). Fig. 10 shows a sample of these datasets. In a handful of cases, data may be missing or corrupt; we have done nothing to rectify this.

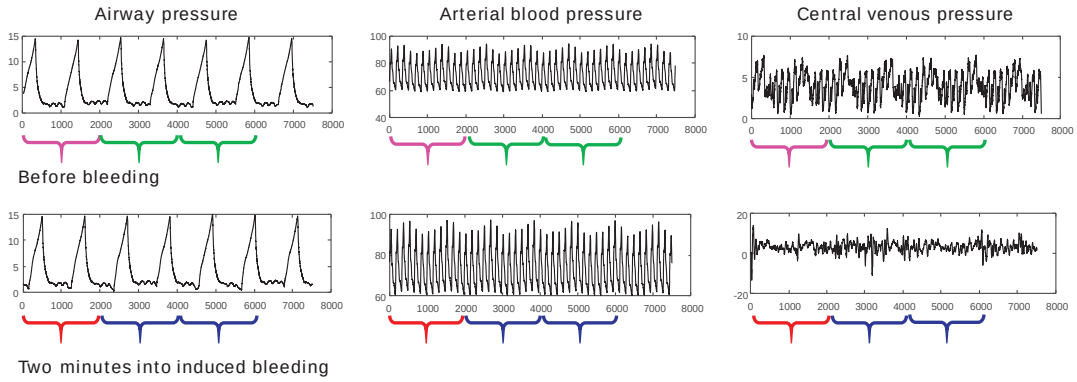

Fig. 10 InternalBleeding datasets. Class i is the i^{th} individual pig. In the training set, class i is represented by two examples, the first 2000 data points of the "*before*" time series (pink braces), and the first 2000 data points of the "*after*" time series (red braces). In the test set, class i is represented by four examples, the second and third 2000 data points of the "before" time series (green braces), and the second and third 2000 data points of the "after" time series (blue braces).

These datasets are interesting and challenging for several reasons. Firstly, the data is not phase aligned, which may call for a phase-invariant or an elastic distance measure. Secondly, the number of classes is huge, especially relative to the number of training instances. Finally, each class is polymorphic, having exactly one example from the pig when it was healthy, and one from when it was injured.

6.3.5 *Electrical Load Measurement data - Freezer datasets*

This dataset was derived from a multi-institution project entitled Personalized Retrofit Decision Support Tools for UK Homes using Smart Home Technology (REFIT) (Murray et al. 2015). The dataset includes data from twenty households from the Loughborough area over the period 2013-2014. All data are from freezers in house 1. This dataset has two classes, one representing the power demand of the fridge freezer

in the kitchen, the other representing the power demand of the (less frequently used) freezer in the garage.

The two classes are difficult to tell apart globally. Each consists of a flat region (the compressor is off), followed by an instantaneous increase (the compressor is turned on), followed by a slower decease as the compressor builds some rotational inertial. Finally, once the temperature has been lowered enough, there is instantaneous fall back to a flat region (this part may be missing in some exemplars). The amount of time the compressor is on can vary greatly, but that is not class dependent. In Fig. 11 however, if you examine the region just after the fiftieth data point, you can see a subtle class conserved difference in how fast the compressor builds rotational inertial and decreases its power demand. An algorithm that can exploit this difference could do well on these datasets, however, global algorithms, such as 1-NN with Euclidean distance may have a hard time beating the default rate.

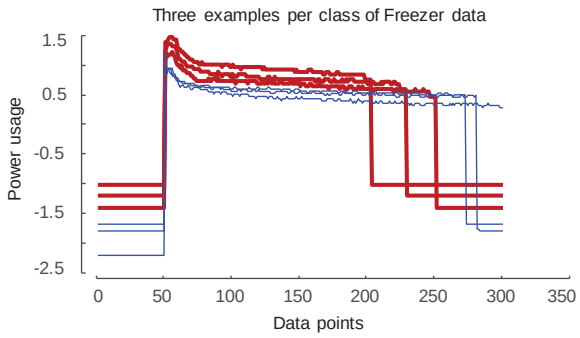

Fig. 11 Some examples from the two-class *Freezer* dataset. The two classes (blue/fine lines vs. red/bold lines) represents the power demand of the fridge freezers sitting in different locations of the house. The classes are difficult to tell apart globally but differ locally.

Freezer is an example of datasets with different train/test splits (the others are *InsectEPG* and *MixedShapes*). We created two train set versions: a smaller train set and a regular train set (both are accompanied by a same test set). This is to meet the community's demand of benchmarking algorithms that are able to work with little train data, for example, generating synthetic time series to augment sparse datasets (Forestier et al. 2017; Dau, Silva, et al. 2018). Some algorithms produce favorable results when training exemplars are abundant but deteriorate when the train data are scarce.

7 Conclusions and Future Work

We have introduced the new 128 datasets version of the UCR Time Series Archive. This resource will be made freely available at the online repository in perpetuity: www.cs.ucr.edu/~eamonn/time_series_data_2018 (Dau, Keogh, et al. 2018). We have further offered advice to the community on best practices on using the archive to test classification algorithms, although we recognize that the community is free to ignore all such advice. Finally, we offered a cautionary tale about how easily practitioners can inadvertently misattribute improvements in classification accuracy. We hope this will encourage all users (including the current authors) to have deeper introspection about the evaluation of proposed distance measures and algorithms.

8 Acknowledgements

We thank the many users of the UCR Time Series Archive, dating back to 2002, who provided us with both success and failure stories, giving us the insight and motivation to produce this work. We also wish to take this opportunity to thank the donors of the data to the archive. Most of the development of the UCR Archive was funded by NSF awards IIS 0803410, 0808770, 0237918 and 1161997. The UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/M015807/1 have also supported this work.

9 References

Agrawal, Rakesh, Christos Faloutsos, and Arun Swami. 1993. “Efficient Similarity Search in Sequence Databases.” In *International Conference on Foundations of Data Organization and Algorithms*, 69–84. Springer.

Bagnall, Anthony, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. “The Great Time Series Classification Bake off: A Review and Experimental Evaluation of Recent Algorithmic Advances.” *Data Mining and Knowledge Discovery* 31 (3): 606–60. https://doi.org/10.1007/s10618-016-0483-9.

Bagnall, Anthony, Jason Lines, William Vickers, and Eamonn Keogh. 2018. “The UEA and UCR Time Series Classification Repository.” www.timeseriesclassification.com.

Batista, Gustavo EAPA, Eamonn J Keogh, Oben Moses Tataw, and Vinicius M A De Souza. 2014. “CID: An Efficient Complexity-Invariant Distance for Time Series.” *Data Mining and Knowledge Discovery* 28 (3). Springer: 634–69.

Benavoli, Alessio, Giorgio Corani, and Francesca Mangili. 2016. “Should We Really Use Post-Hoc Tests Based on Mean-Ranks.” *Journal of Machine Learning Research* 17 (5): 1–10.

Brodersen, Kay Henning, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. “The Balanced

Accuracy and Its Posterior Distribution." In *Pattern Recognition (ICPR), 2010 20th International Conference On*, 3121–24. IEEE.

Chen, Yanping, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. 2015. "The UCR Time Series Classification Archive." www. cs. ucr. edu/~ eamonn/time_series_data.

Dau, Hoang Anh, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, et al. 2018. "The UCR Time Series Classification Archive." https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Dau, Hoang Anh, Diego Furtado Silva, François Petitjean, Germain Forestier, Anthony Bagnall, and Eamonn Keogh. 2017. "Judicious Setting of Dynamic Time Warping's Window Width Allows More Accurate Classification of Time Series." In *IEEE International Conference on Big Data (Big Data)*, 917–22. http://ieeexplore.ieee.org/document/8258009/.

Dau, Hoang Anh, Diego Furtado Silva, Francois Petitjean, Germain Forestier, Anthony Bagnall, Abdullah Mueen, and Eamonn Keogh. 2018. "Optimizing Dynamic Time Warping's Window Width for Time Series Data Mining Applications." *Data Mining and Knowledge Discovery*. https://doi.org/10.1007/s10618-018-0565-y.

Demšar, Janez. 2006. "Statistical Comparisons of Classifiers over Multiple Data Sets." *Journal of Machine Learning Research* 7 (Jan): 1–30.

Fisher, Ronald A. 1936. "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7 (2). Wiley Online Library: 179–88.

Forestier, Germain, Francois Petitjean, Hoang Anh Dau, Geoffrey I. Webb, and Eamonn Keogh. 2017. "Generating Synthetic Time Series to Augment Sparse Datasets." In *2017 IEEE International Conference on Data Mining (ICDM)*, 865–70. https://doi.org/10.1109/ICDM.2017.106.

Friedman, Milton. 1937. "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance." *Journal of the American Statistical Association* 32 (200). Taylor & Francis: 675–701.

———. 1940. "A Comparison of Alternative Tests of Significance for the Problem of m Rankings." *The Annals of Mathematical Statistics* 11 (1). JSTOR: 86–92.

Garcia, Salvador, and Francisco Herrera. 2008. "An Extension On``statistical Comparisons of Classifiers over Multiple Data Sets''for All Pairwise Comparisons." *Journal of Machine Learning Research* 9 (Dec): 2677–94.

Gharghabi, Shaghayegh, Shima Imani, Anthony Bagnall, Amirali Darvishzadeh, and Eamonn Keogh. 2018. "Matrix Profile XII: MPdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios." In *To Appear in ICDM 2018*.

Guillame-Bert, Mathieu, and Artur Dubrawski. 2017. "Classification of Time Sequences Using Graphs of Temporal Constraints." *Journal of Machine Learning Research* 18: 1–34.

He, Yuanduo, Jialiang Pei, Xu Chu, Yasha Wang, Zhu Jin, and Guangju Peng. 2018. "Time Series Classification via Manifold Partition Learning." In *To Appear in ICDM 2018*.

Hills, Jon, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. 2014. "Classification of Time Series by Shapelet Transformation." *Data Mining and Knowledge Discovery* 28 (4). Springer: 851–81.

Hollander, Myles, and Douglas A Wolfe. 1999. "Nonparametric Statistical Methods." Wiley-Interscience.

Holm, Sture. 1979. "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics*.

JSTOR, 65–70.

Hu, Bing, Yanping Chen, and Eamonn Keogh. 2016. “Classification of Streaming Time Series under More Realistic Assumptions.” *Data Mining and Knowledge Discovery* 30 (2): 403–37. https://doi.org/10.1007/s10618-015-0415-0.

Huang, Yun-Wu, and Philip S Yu. 1999. “Adaptive Query Processing for Time-Series Data.” In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 282–86. ACM.

Keogh, Eamonn, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. “Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases.” *Knowledge and Information Systems* 3 (3). Springer: 263–86.

Keogh, Eamonn, and T Folias. 2002. “The UCR Time Series Data Mining Archive [Http://Www. Cs. Ucr. Edu/~ Eamonn/TSDMA/Index. Html]. Riverside CA.” *University of California-Computer Science & Engineering Department*.

Keogh, Eamonn, and Shruti Kasetty. 2003. “On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration.” In *Data Mining and Knowledge Discovery*, 7:349–71. https://doi.org/10.1023/A:1024988512476.

Kim, Edward D, Joyce M W Lam, and Jiawei Han. 2000. “Aim: Approximate Intelligent Matching for Time Series Data.” In *International Conference on Data Warehousing and Knowledge Discovery*, 347–57. Springer.

Li, Daoyuan, Tegawende F Bissyande, Jacques Klein, and Yves Le Traon. 2016. “Time Series Classification with Discrete Wavelet Transformed Data.” *International Journal of Software Engineering and Knowledge Engineering* 26 (09n10). World Scientific: 1361–77.

Li, Daoyuan, Tegawendé François D Assise Bissyande, Jacques Klein, and Yves Le Traon. 2016. “Time Series Classification with Discrete Wavelet Transformed Data: Insights from an Empirical Study.” In *The 28th International Conference on Software Engineering and Knowledge Engineering (SEKE 2016)*.

Lichman, Moshe. 2013. “UCI Machine Learning Repository.”

Lines, Jason, Sarah Taylor, and Anthony Bagnall. 2016. “Hive-Cote: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification.” In *Data Mining (ICDM), 2016 IEEE 16th International Conference On*, 1041–46. IEEE.

———. 2018. “Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles.” *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12 (5). ACM: 52.

Lipton, Zachary C, and Jacob Steinhardt. 2018. “Troubling Trends in Machine Learning Scholarship.” *ArXiv Preprint ArXiv:1807.03341*.

Lu, Sha, Gordana Mirchevska, Sayali S. Phatak, Dongmei Li, Janos Luka, Richard A. Calderone, and William A. Fonzi. 2017. “Dynamic Time Warping Assessment of Highresolution Melt Curves Provides a Robust Metric for Fungal Identification.” *PLoS ONE* 12 (3). https://doi.org/10.1371/journal.pone.0173320.

Mezari, Antigoni, and Ilias Maglogiannis. 2017. “Gesture Recognition Using Symbolic Aggregate Approximation and Dynamic Time Warping on Motion Data.” In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 342–47. ACM.

Murray, David, Jing Liao, Lina Stankovic, Vladimir Stankovic, Charlie Wilson, Michael Coleman, and Tom Kane. 2015. “A Data Management Platform for Personalised Real-Time Energy Feedback.” *Eedal*, 1–15.

https://pure.strath.ac.uk/portal/files/45214811/Murray_etal_EEDAL_2015_A_data_management_platform_for_personalised_real_time.pdf.

Okeh, U M, and C N Okoro. 2012. “Evaluating Measures of Indicators of Diagnostic Test Performance: Fundamental Meanings and Formulars.” *J Biom Biostat* 3 (1): 2.

Paparrizos, John, and Luis Gravano. 2015. “K-Shape: Efficient and Accurate Clustering of Time Series.” *Acm Sigmod*, 1855–70. https://doi.org/10.1145/2723372.2737793.

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2013. “Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping.” *Transactions on Knowledge Discovery from Data (TKDD*. https://doi.org/10.1145/2500489.

Saito, Naoki, and Ronald R Coifman. 1995. “Local Discriminant Bases and Their Applications.” *Journal of Mathematical Imaging and Vision* 5 (4). Springer: 337–58.

Salzberg, Steven L. 1997. “On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach.” *Data Mining and Knowledge Discovery* 1 (3). Springer: 317–28.

Schäfer, Patrick. 2015. “The BOSS Is Concerned with Time Series Classification in the Presence of Noise.” *Data Mining and Knowledge Discovery* 29 (6). Springer: 1505–30.

Siegal, Sidney. 1956. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-hill.

Silva, Angelo, and Renato Ishii. 2016. “A New Time Series Classification Approach Based on Recurrence Quantification Analysis and Gabor Filter.” In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 955–57. ACM.

Silva, Diego F., Gustavo E.A.P.A. Batista, and Eamonn Keogh. 2017. “Prefix and Suffix Invariant Dynamic Time Warping.” In *Proceedings - IEEE International Conference on Data Mining, ICDM*, 1209–14. https://doi.org/10.1109/ICDM.2016.107.

Student. 1908. “The Probable Error of a Mean.” *Biometrika*. JSTOR, 1–25.

Taktak, Mariem, Slim Triki, and Anas Kamoun. 2017. “SAX-Based Representation with Longest Common Subsequence Dissimilarity Measure for Time Series Data Classification.” In *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference On*, 821–28. IEEE.

Uguroglu, Selen. 2013. “Robust Learning with Highly Skewed Category Distributions.” PhD thesis, Carnegie Mellon University.

Wilcoxon, Frank. 1945. “Individual Comparisons by Ranking Methods.” *Biometrics Bulletin* 1 (6). JSTOR: 80–83.

Zhang, Hui, and Tu Bao Ho. 2005. “Finding The Clustering Consensus of Time Series with Multi-Scale Transform.” In *Soft Computing as Transdisciplinary Science and Technology*, 1081–90. Springer.

Zhang, Hui, Tu Bao Ho, Mao Song Lin, and Wei Huang. 2006. “Combining the Global and Partial Information for Distance-Based Time Series Classification and Clustering.” *JACIII* 10 (1): 69–76.