Should we Reload Time Series Classification Performance Evaluation ? (a position paper)

Dominique Gay[1] and Vincent Lemaire[2]

[1] Laboratoire d'Informatique et de Mathématiques EA2525
Université de La Réunion, La Réunion, France
[2] Orange Labs
Lannion, France

Abstract. Since the introduction and the public availability of the UCR time series benchmark data sets, numerous Time Series Classification (TSC) methods has been designed, evaluated and compared to each others. We suggest a critical view of TSC performance evaluation protocols put in place in recent TSC literature. The main goal of this "position" paper is to stimulate discussion and reflexion about performance evaluation in TSC literature.

1 Introduction

The *need for time series data mining benchmarks* [15] has been fulfilled: with, firstly, the UCR Time Series Classification Archive [9] and then the UEA & UCR Time Series Classification Repository [5], the research community now have more than 85 data sets from various application domains to evaluate newly introduced TSC methods and to compare them to existing ones. The public availability and the wide diffusion of the benchmark data had a strong and positive impact on the research community which has been prolific in terms of publications of TSC methods; as an example, the recent experimental evaluation in [3] involves 18 recently proposed algorithms while in the same year, in 2017, two contenders, BoPF [17] and WEASEL [23] have been presented in top data mining conferences.

In most of the research papers, the experimental evaluation section starts with *"Each UCR dataset provides a train and test split set which we use unchanged to make our results comparable the prior publications"*[3]. And this is where the reflexion we suggest begins. In the following, we will consider accuracy as the measure of predictive performance since it is the most widely used in TSC literature.

[3] quoted from WEASEL paper [23].

2 Discussion & reflexion

About train/test experiment.
While in transactional data classification, resampling strategies (mainly bootstrap and stratified k-fold cross-validation) are the norm to estimate the expected performance of classifiers and to compare them [22, 11], there is a singularity concerning predictive performance evaluation in TSC literature: the vast majority of research work restrains predictive performance evaluation to a single train/test split experiment [5], also called hold-out method or test sample estimation [16].

Unless disposing of a *large and representative* data set of the application domain, a single train/test experiment is generally ineffective in providing predictive performance estimation and valuable comparison between TS classifiers without random subsamplings (i.e., repeated hold out experiments). Indeed, different samplings may lead to results with strong variations [16]. Thus, a classifier $\mathcal{C}_A$ may show better predictive performance than another classifier $\mathcal{C}_B$ just because of this particular split. And, in such cases, subsequent statistical tests based on single train/test accuracy results (such as now commonly used post-hoc Nemenyi test [12]) do not help more in comparing classifiers performance over several data sets since other train/test splits could have led to different accuracy and mean rank results and thus potentially different conclusions.

Moreover, unless disposing of tens of thousands instances, it is common to keep more instances in the training set than in the test hold out, generally around class-stratified ratio of 2/3 for training against the rest i.e., 1/3 for testing. The TSC repository [5] provides various predefined train/test split ratios, going from 1.6% to 81% of the whole data set for training set (notice that 35 of the 85 TSC data sets show a train/test split ratio below 34%). In addition to small –if not very small– training sets, the train/test splits are not always class-stratified; it results in very (too) few representative instances of some class labels (especially in multi-class problems). We are aware that, in some application domains, "labelled data is expensive to collect" [3], however, considering the whole available and *already-labelled* data, splitting for such small non-stratified training sets is also questionable. Here are some singularities that arise from some data sets drawn from the TSC archive:

- DiatomSizeReduction: a 4-class problem, with a train/test split ratio about 5% (16 instances for training, 306 for testing) and a class distribution (1,6,5,4) for training against (33,92,94,87) for testing. The class distribution is not respected from train to test set. There are relatively 73% more instances of class c_1 in the test set than in the training set.
- SonyAIRobotSurface1: a 2-class problem, with a train/test split ratio about 3.22% (20 instances for training, 601 for testing) and a class distribution (6,14) for training against (343,258) for testing. The class distribution is not respected from train to test set. There are relatively 90% more instances of class c_1 in the test set than in the training set; and 38% less instances of class c_2 in the test set than in the training set.
- ...several additional similar cases of data sets can be found in commonly used benchmark [5] : either the number of training examples is very small

w.r.t the whole available data, either there is class distribution change between training set and test set (sometimes both singularities arise).

Generally speaking, in addition to the weakness of single train/test experiments for performance evaluation, choosing to split such way, without class-stratification in small training sets, results in "unnecessary difficult" TSC problems: indeed, the obtained training set is not always representative of the whole available data set and as explained above, it could lead to class distribution change between training and test sets (also known as prior probability shift [20]). In such environments, intuitively, simple 1-Nearest Neighbor lazy learners [27] and ensemble methods that embed several 1-NN classifiers (with various distance measures or on various data representations) [2, 4, 18] still obtain "good" accuracy results since it is still possible to find a nearest neighbor of a test instance in very few training instances of a minor class. However, eager classifiers based on empirically observed per-class frequencies (such as Naïve Bayes or Decision Trees) often fail in characterizing the minor classes with very few representative instances.

If the TSC archive [5] offers a wide variety of data sets, the original train/test splits also involves hidden difficult well-known problems in the Machine Learning community: e.g., learning from few examples or in class distribution change environment. Unfortunately, averaging ranks over all data sets to produce critical difference diagram presents a risk of hiding the reasons why a particular classifier shows better performance than another. As an example [4], in Table 1, we report performance comparisons of 11 recent classifiers like in recent literature (single train/test split following by significance testing). We provide several statistical tests by integrating step by step data sets which involves smaller size of training set.

Min. size	#DB	CD	WEASEL	DTWCV	DTW	BOSS	LS	TSBF	ST	EE	CoTE	SNB	BoPF
$\geq$ 1000	10	*4.77403*	**3.250**	8.300	8.600	**5.700**	**6.050**	**7.700**	**3.050**	**6.250**	**4.100**	**3.400**	9.600
> 500	22	*3.21865*	**2.932**	7.659	9.023	**5.705**	6.909	6.841	**4.068**	6.273	**3.636**	**4.000**	8.955
> 300	42	*2.32949*	**3.560**	7.929	8.893	5.857	6.845	6.298	**4.500**	6.060	**3.369**	**4.821**	7.869
> 200	48	*2.17903*	**3.594**	7.917	8.885	5.781	6.885	6.271	**4.625**	5.865	**3.219**	**5.010**	7.948
> 100	57	*1.99962*	**3.737**	7.868	8.860	5.474	6.623	6.237	**4.544**	5.930	**3.298**	5.535	7.895
All	85	*1.63748*	**3.847**	7.806	8.647	5.382	6.135	6.388	**4.847**	5.871	**3.412**	6.429	7.235

Table 1. Average ranks of 11 recent classifiers depending on the data sets taken into account from [5] (i.e., depending on the training set size). Accuracy results are taken from Schäfer & Leser paper on WEASEL [23]. Post-hoc Nemenyi's statistical test considering training set with size ≥ 1000 (only 10 data sets), then > 500 (only 22 data sets), . . . , until considering All 85 data sets from [5]. Underlined rank is the best per line and bold results on the same line indicates that there is no statistically significant difference of performance between bold results according to Nemenyi's test, considering the current benchmark datasets.

[4] We did not integrate recently introduced HIVE-COTE [19] accuracy results since they are available under 100 resamples protocol.

WEASEL and ST [14, 7] score the highest mean ranks when considering data sets with training set size greater than 500 while CoTE takes advantage when adding data with smaller training size. We also observe that the mean rank of ST increases as we consider more and more data sets with smaller training size. Notice that, it has to be balanced against the fact that as the number of data sets decreases the critical difference (CD) value increases, making more difficult to state significant differences of performance between contenders. Another illustrative example is about SNB [8] (an improved Naïve Bayes classifier benefiting from multiple representations) which is competitive with WEASEL, CoTE, BOSS and ST when not considering too small (< 200) training set size – confirming the importance of the size of the training set on the predictive performance of some classifiers. Aside from [27], we may regret the lack of experimental studies about the learning curves [21, 24] of TSC algorithms.

About resampling strategies.
As far as we know, very few attempts of resampling strategies for TSC performance evaluation have been led, e.g., :

- Grabocka et al. [13] provides some results by 5-folds cross validation on 35 UCR data sets. It allows to identify easy data sets in the TSC archive [9]. Indeed, a default SVM classifier with polynomial kernel scores above 95% accuracy (often near perfect) on 18 data sets.
- Wang et al. [27], focusing on 1-NN with various distance measures, provides k-folds stratified cross-validation results. However, the k varies from 2 to 30 depending on the benchmark data set and the cross-validation method at use is unconventional: when splitting the data set T into k folds, the model is learnt on fold T_k and tested on $T \setminus T_k$ –while conventional k-folds cross validation being the opposite: learning the model on $T \setminus T_k$ and testing on T_k. This unconventional cross-validation leads to the same problems explained above (with 1/30, i.e., 3% of the whole data set used for training).
  In the same paper, the authors also noticed the importance of the effect of the training data set size on 1-NN classifier accuracy : DTW-1-NN is better than ED-1-NN with small training data set, but providing that we have enough training instances (a few hundred/thousand depending on the simulated data set), the two classifiers show similar predictive performance.
- Bagnall et al. [3] performs 100 resampling experiments on each of the 85 TSC data sets –followed by Nemenyi's statistical post-hoc test. However, the multiple resamplings fit the original size of train/test split provided by [5] – which leads to the same problem raised above about training size and class distribution change. This resampling strategy gives, all the same, a better idea of the performance of recent classifiers on data with *various* sizes of training sets, although the train/test split is still questionable. Indeed, if instances from original test set are authorized to be in training set due

to resampling, why not use 10-CV or 10×10 CV, as in transactional data classification literature[5] ?

The cross-validation (CV) method is not unknown to the TSC community; indeed, some algorithms, like e.g., WEASEL or DTW-CV used cross-validation on training set to set hyper-parameters (even if the training set is very small), then a single train/test split is performed to evaluate the performance of the "best hyper-parametered" model on a single hold out test set. Again, why not use CV method to evaluate and compare TSC algorithms ?

About (repeated) 10-folds CV, statistical tests and beyond.
While 10-folds CV with subsequent statistical tests [12] is now the gold standard for predictive performance evaluation and comparisons between classifiers on transactional benchmark data sets, recently, Vanwinckelen & Blockeel [25, 26] warns the Machine Learning community about pitfalls hidden in such comparisons. The take away messages are:

- *"Our experiments show that when using cross-validation for choosing between two models, the best performing model is not always chosen".*
- *"This discussion leads us to question the usefulness of statistical testing in the context of evaluating predictive models with cross-validation".*

On the other hand, after almost a decade of the *"10-CV + statistical tests"* combination [12] to evaluate learner's predictive performance, J. Demsar et al. [6, 10] "*discourage the use of frequentist null hypothesis significance tests (NHST) in machine learning and, in particular, for comparison of the performance of classifiers*" and encourages the community to embrace Bayesian analysis using 10×10-CV for comparing classifiers. This is perhaps a change point for performance evaluation habits in the Machine Learning community. Notice that Bayesian analysis of performance is more conservative than NHST; that is, it is "more difficult" for a classifier $\mathcal{C}_A$ to be better than a classifier $\mathcal{C}_B$, considering Bayesian analysis.

About the evaluation measure.
As recalled in the introduction, the vast majority of recently proposed TSC algorithms are evaluated and compared with regards to the accuracy measure, i.e., the number of correctly classified time series. Accuracy measure is suitable for roughly balanced 2-class data sets. However, for unbalanced and/or multiclass data sets, accuracy measure is inappropriate for evaluation since high accuracy results due to a bias towards the majority class could hide severely bad predictive performance on the minor class or on other classes in multiclass settings. Often, ROC or Precision/Recall curve analysis or lift curve and cumulative gain charts are prefered for unbalanced settings.

The TSC archive [5] contains some 2-class unbalanced problems, e.g., Earthquakes, ToeSegmentation2 and Wafer with respectively 20.2%, 25.3% and 10.6%

[5] However, even with k-folds CV, a particular attention must be given to the setting of k for the 17 small data sets, with less than 200 instances, from the TSC archive.

unbalanced ratio (i.e., the proportion of the minor class). The archive also contains many multiclass problems with severe unbalanced ratios, e.g., ECG5000 and Worms with respectively 0.5% and 9.7% unbalanced ratio. Learning in the presence of class imbalance or in multiclass settings is still an ongoing machine learning research topic [1]. Again, the presence of such data sets in the benchmark repository, when averaging ranks of classifiers based on accuracy results, could lead to flawed conclusions on the performance evaluation.

3 Conclusion

The still ongoing public release of benchmark TSC data sets to the data mining community through the UCR & UEA repository has certainly been the best catalyst for the development of new TSC algorithms and methods. In this discussion paper, we briefly review the habitual protocols at use in TSC algorithms performance evaluation, discuss the pros and cons of experimental protocols and try to warn the community about the pitfalls and hidden problems of actual performance evaluation protocols in TSC literature. We agree that the core of this discussion paper needs more in-depth development and experimental arguments but we believe that interesting discussions on this important topic deserve to be launched and continued during the 3^{rd} ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data.

Acknowledgments

We would like to thank the anonymous reviewers for joining the discussion with valuable comments and arguments.

References

1. Proceedings of the IJCAI 2017 Workshop on Learning in the Presence of Class Imbalance and Concept Drift (LPCICD'17) (2017)
2. Bagnall, A., Davis, L.M., Hills, J., Lines, J.: Transformation based ensembles for time series classification. In: Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012. pp. 307–318 (2012)
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.J.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov. 31(3), 606–660 (2017)
4. Bagnall, A., Lines, J., Hills, J., Bostrom, A.: Time-series classification with COTE: the collective of transformation-based ensembles. IEEE Trans. Knowl. Data Eng. 27(9), 2522–2535 (2015)
5. Bagnall, A., Lines, J., Vickers, W., Keogh, E.: The UEA & UCR time series classification repository, `www.http://timeseriesclassification.com`

6. Benavoli, A., Corani, G., Demsar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. Journal of Machine Learning Research 18, 77:1–77:36 (2017)
7. Bostrom, A., Bagnall, A.: Binary shapelet transform for multiclass time series classification. In: Big Data Analytics and Knowledge Discovery - 17th International Conference, DaWaK 2015, Valencia, Spain, September 1-4, 2015, Proceedings. pp. 257–269 (2015)
8. Boullé, M.: Towards automatic feature construction for supervised classification. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I. pp. 181–196 (2014)
9. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (July 2015), `www.cs.ucr.edu/~eamonn/time_series_data/`
10. Corani, G., Benavoli, A., Demsar, J., Mangili, F., Zaffalon, M.: Statistical comparison of classifiers through bayesian hierarchical modelling. Machine Learning 106(11), 1817–1837 (2017)
11. Delgado, M.F., Cernadas, E., Barro, S., Amorim, D.G.: Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research 15(1), 3133–3181 (2014)
12. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Grabocka, J., Nanopoulos, A., Schmidt-Thieme, L.: Invariant time-series classification. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II. pp. 725–740 (2012)
14. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Min. Knowl. Discov. 28(4), 851–881 (2014)
15. Keogh, E.J., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Min. Knowl. Discov. 7(4), 349–371 (2003)
16. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes. pp. 1137–1145 (1995)
17. Li, X., Lin, J.: Linear time complexity time series classification with bag-of-pattern-features. In: 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017. pp. 277–286 (2017)
18. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Min. Knowl. Discov. 29(3), 565–592 (2015)
19. Lines, J., Taylor, S., Bagnall, A.J.: HIVE-COTE: the hierarchical vote collective of transformation-based ensembles for time series classification. In: IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. pp. 1041–1046 (2016)
20. Moreno-Torres, J.G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition 45(1), 521–530 (2012)
21. Perlich, C., Provost, F.J., Simonoff, J.S.: Tree induction vs. logistic regression: A learning-curve analysis. Journal of Machine Learning Research 4, 211–255 (2003)
22. Salzberg, S.: On comparing classifiers: Pitfalls to avoid and a recommended approach. Data Mining & Knowledge Discovery 1(3), 317–328 (1997)

23. Schäfer, P., Leser, U.: Fast and accurate time series classification with WEASEL. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017. pp. 637–646 (2017)
24. Ting, K.M., Washio, T., Wells, J.R., Aryal, S.: Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. Machine Learning 106(1), 55–91 (2017)
25. Vanwinckelen, G., Blockeel, H.: On estimating model accuracy with repeated cross-validation. In: Proceedings BeneLearn'2012 (2012)
26. Vanwinckelen, G., Blockeel, H.: Look before you leap: Some insights into learner evaluation with cross-validation. In: 1st ECML/PKDD Workshop on Statistically Sound Data Mining, SSDM 2014, held at ECML/PKDD 2014, Nancy, France, September 15, 2014. pp. 3–20 (2014)
27. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.J.: Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Discov. 26(2), 275–309 (2013)