Knowl Inf Syst (2018) 55:493–528
https://doi.org/10.1007/s10115-017-1090-9

REGULAR PAPER

# Self-labeling techniques for semi-supervised time series classification: an empirical study

**Mabel González**[1] · **Christoph Bergmeir**[2] · **Isaac Triguero**[3] · **Yanet Rodríguez**[1] · **José M. Benítez**[4]

Received: 19 May 2016 / Revised: 14 July 2017 / Accepted: 28 July 2017 /
Published online: 8 August 2017


**Abstract** An increasing amount of unlabeled time series data available render the semi-supervised paradigm a suitable approach to tackle classification problems with a reduced quantity of labeled data. Self-labeled techniques stand out from semi-supervised classification methods due to their simplicity and the lack of strong assumptions about the distribution of the labeled and unlabeled data. This paper addresses the relevance of these techniques in the time series classification context by means of an empirical study that compares successful self-labeled methods in conjunction with various learning schemes and dissimilarity measures. Our experiments involve 35 time series datasets with different ratios of labeled data, aiming to measure the transductive and inductive classification capabilities of the self-labeled methods studied. The results show that the nearest-neighbor rule is a robust choice for the base classifier. In addition, the amending and multi-classifier self-labeled-based approaches reveal a promising attempt to perform semi-supervised classification in the time series context.

**Keywords** Semi-supervised classification · Self-labeled · Time series classification · Semi-supervised learning · Self-training

## 1 Introduction

In the time series field, the semi-supervised learning (SSL [12]) paradigm has received a lot of research attention during the past decade. As cheap sensors of all kinds become more

✉ Christoph Bergmeir
christoph.bergmeir@monash.edu

1 Department of Computer Science, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba

2 Faculty of Information Technology, Monash University, P.O. Box 63, Melbourne, VIC 3800, Australia

3 School of Computer Science, University of Nottingham, Nottingham, UK

4 Department of Computer Science and Artificial Intelligence, E.T.S. de Ingenierías Informática y de Telecomunicación, University of Granada, Granada, Spain

and more available, vast amounts of unlabeled time series data are generated. By contrast, the cost related to the labeling process makes it often unfeasible to obtain a fully labeled training set. In this situation, SSL is a good solution to improve the learning accuracy taking advantage of the unlabeled data in conjunction with a small set of labeled data. Specifically, semi-supervised classification (SSC) focuses on training a classifier such that it outperforms a supervised classifier trained on the labeled data alone. In the time series domain, SSL has found a wide range of applications such as the classification of flying insects [5], web information extraction [23] and failure prediction in oil production [40]. In this work, we tackle SSC problems in the time series classification context.

Time series data are characterized by a high dimensionality and its numerical and continuous nature. Therefore, a special treatment must be considered to deal with time series classification [25]. A first category of proposals, called *feature-based* approach [7,11,17,26,62], transforms the original time series into a new description space where conventional classifiers can be applied. Signal processing or statistical tools are commonly used to extract features from the original raw data. A second category [21,34,46,52,53,64] focuses on customizing or developing classifiers specifically designed for time series data. This category, which includes the *instance-based* approach, is mostly based on the selection of an appropriate representation of the time series and a suitable measure of dissimilarity. Several representations and dissimilarity measures have been proposed to deal with the time series classification problem including: spectral approaches [22], autocorrelation function [2] and elastic measures [13,41,54]. Our paper considers this second approach.

Several SSC approaches have been applied to the time series classification problem. The work of Marussy and Buza [43] uses the cluster-then-label approach by a constrained hierarchical single link clustering method. The work of Frank et al. [24] applies a similar approach with a similarity measure called geometric template matching. The applicability of graph-based methods to time series classification is addressed in various works [18,19]. The classical semi-supervised support vector machines method is extended to tackle time series classification by Kim [35].

Another family of SSC methods, denoted self-labeled techniques [57], aims to enlarge the original labeled set using the most confident predictions to classify unlabeled data. In contrast to the previous mentioned approaches, self-labeled techniques do not make any special assumptions about the distribution of the input data. Self-training [67] and co-training [9] are the most popular self-labeled techniques. Both approaches have been applied in a time series context.

Self-training is a wrapper method that iteratively classifies the unlabeled examples, assuming that its more accurate predictions tend to be correct. In the time series domain, the self-training technique is mostly applied to a particular case of SSC, called positive unlabeled learning, within which only examples from one class are available [6,14,30,51,61]. In conjunction with the self-training, the $k$-nearest-neighbor ($k$NN [1]) is typically used as the base learner as it has shown to be particularly effective for time series classification tasks [55,58].

Co-training is a SSC method that requires two conditionally independent views which are sufficient for learning and classification. For each view, the unlabeled instances with highest confidence are selected and labeled to be turned into additional training data for the other view. The multi-view requirement of the co-training technique is typically too strong and difficult to meet in the time series domain where in most cases observations close together are correlated. The work of Meng et al. [44] applies a variant of co-training [29] which uses the hidden Markov model [49] and one-nearest-neighbor (1NN) as two different learners instead of the classical two views of the data.

There are various works in the literature [6,14,44,51,61] that focus on the SSC of time series, which involve self-labeled techniques. Self-training and co-training are the only self-labeled techniques that have been applied in a time series context so far, to the best of our knowledge. Our study broadens this approach to other self-labeled techniques, therewith gaining new insights and allowing for more detailed conclusions on this topic. Moreover, despite the successful application of 1NN to time series classification tasks, the use of different learning approaches as a base classifiers seems to be an under-explored area. These reasons motivate our paper, which has three main objectives as follows:

- To explore the applicability of classical self-labeled techniques in the time series domain, as well as the use of other classification schemes as base learners in addition to the well-known 1NN.
- To identify the best methods for each base learner under different ratios of labeled data and dissimilarity measures.
- To determine the influence of the geometrical characteristics of time series datasets in the performance of the self-labeled techniques.

The rest of the paper is organized as follows. In Sect. 2, we provide definitions and the notation of SSC in the time series domain. Furthermore, we discuss the main characteristics of the self-labeled techniques and the base learners involved in this study. In Sect. 3, we introduce the experimental framework. In Sect. 4, we present the results obtained and discuss them. Finally, Sect. 5 concludes the paper.

## 2 Semi-supervised time series classification

In this section, we define the SSC of time series problem and the principal notation and definitions. Furthermore, we review the self-labeled techniques and the base learner methods involved in this study (Sects. 2.1, 2.2).

In SSC, the source dataset has two parts, $L$ and $U$. Let $L$ be the set of instances $\{x_1, \ldots, x_l\}$ for which the labels $\{y_1, \ldots, y_l\}$ are provided. Let $U$ be the set of instances $\{x_{l+1}, \ldots, x_{l+u}\}$ for which the labels are not known. We follow the typical assumption that there is much more unlabeled than labeled data, i.e., $u \gg l$. The whole set $L \cup U$ forms the training set.

Throughout this study, each instance $x_i, i = 1, \ldots, l + u$, represents a univariate, real-valued and evenly spaced time series. In this case, the time series $x_i = [p_1, p_2, \ldots, p_n]$ is considered a sequence of $1$-dimensional data points.

Depending on the goal, we can categorize SSC into two slightly different settings [12], namely transductive and inductive learning. The former is devoted to predict the labels on the unlabeled instances $U$ provided during the training phase. The latter aims to predict the labels on unseen testing instances. In this work, we delve into both settings aiming to perform an extensive analysis of the selected methods.

### 2.1 Self-labeled techniques

Self-labeled techniques follow a wrapper methodology using one or more supervised classifiers to determine the most likely class of the unlabeled instances during an iterative process. The base classifier(s) play(s) a crucial role in the estimation of the most confident instances of $U$. The main feature that distinguishes self-labeled methods is the way they obtain one or several enlarged labeled sets to efficiently represent the learned hypothesis from the training

set. In the literature, there are several proposals that follow this approach which differ mainly in the following aspects:

- *Addition mechanism* There is a variety of schemes in which the enlarged set can be formed. The most used ones are *incremental* and *amending*. The former adds, step-by-step, the most confident instances from $U$ to the enlarged set. The latter allows rectifications to already labeled instances to avoid the introduction of noisy instances in the enlarged set(s).
- *Classification model* Self-labeled techniques can use one or more base classifiers to establish the class of unlabeled instances. *Single-classifier* models assign to each instance the most likely class considering the used classifier. *Multi-classifier* models combine the hypothesis learned by several classifiers to estimate the class by agreement of classifiers or combination of the probabilities obtained by single-classifiers.
- *Learning approach* Independently of the number of base classifiers, the number of learning methods is another important issue. The *single learning* approach can be linked with single- and multi-classifier models. By contrast, the *multi-learning* approach is intrinsically related to the multi-classifier model. In a multi-learning method, the different classifiers come from different learning methods.
- *Stopping criterion* This is the mechanism used to stop the self-labeling process preventing the addition of labeled instances in $L$ with a low confidence level. Often, a prefixed number of iterations is used as stopping criterion. Another criterion used is the occurrence of non-changes in the learned hypotheses during successive iterations of the self-training process.

Since each approach has its own benefits and drawbacks, we include in this study a representative sample of methods. The selection made is based on the results obtained in the extensive overview study of Triguero et al. [57], and it includes the following methods:

- Standard self-training (SelfT [67]): is a single-classifier and single learning method that extends the set $L$ with the most confident examples extracted and classified from $U$, during an iteratively and incremental process. The stopping criterion consists in a fixed number of iterations that can be adapted to the original size of $U$. Following a wrapper methodology, the base classifier used by self-training is considered as another parameter of the method.
- Self-training with editing (SETRED [38]): is a self-training variant with a different addition mechanism. SETRED introduces a data editing method to filter the noise examples that has been labeled by the base classifier. For each iteration, the mislabeled examples are identified using the local information provided by the neighborhood graph [71].
- Self-training nearest-neighbor rule using cut edges (SNNRCE [59]): is a variant of SETRED that includes a first stage where the most confident examples are added to $L$. In the next stage, the self-training standard is applied in combination with the 1NN rule as a base classifier. The iterative process stops when the expected number of examples in the minority class is reached, according to the distribution observed in $L$. In the final stage, the mislabeled examples are relabeled attending to the information provided by the neighborhood graph.
- Tri-training (TriT [70]): is a variant of co-training that trains three instead of the traditional two classifiers. These classifiers have in common the same learning scheme. The diversity between the base classifiers is obtained through manipulating the original set $L$, for example, using Bagging [10]. For each iteration, the selected examples from $U$ are labeled and added to the training set of a specific classifier only if there is agreement between the remaining classifiers and some conditions are satisfied. The stopping criterion is reached

**Table 1** Summary of the self-labeled methods selected

| Abbreviated name | Addition mechanism | Classification model | Learning approach | Time iteration complexity |
|---|---|---|---|---|
| SelfT | Incremental | Single | Single | $T_c(u) + T_t(l + l')$ |
| SETRED | Amending | Single | Single | $T_c(u) + O((l + l')^3) + T_t(l + l'')$ |
| SNNRCE | Amending | Single | Single | $T_c(u) + T_t(l + l')$ |
| TriT | Incremental | Multi | Single | $6T_c(u) + \sum_{i=1}^{3} T_t(l_i + l'_i)$ |
| Democratic | Incremental | Multi | Multi | $\sum_{i=1}^{N} T_c^i(u) + \sum_{i=1}^{N} T_t^i(l_i + l'_i)$ |

when the hypothesis of the three classifiers does not suffer any modification during a complete iteration.

– Democratic co-learning (Democratic [69]): is a multi-classifier and multi-learning method. The specific number of classifiers and its learning scheme are established as arguments of the method. Initially, all classifiers are trained using the examples in $L$. For each iteration, a label for each unlabeled example is proposed via majority vote. If the classification provided by a classifier disagrees with most classifiers, in a particular example, then this example is included in the training set of the classifier. The iteratively process stops when the training sets of the classifiers do not suffer any additions during a complete iteration.

Table 1 summarizes the principal characteristics of the self-labeled methods selected.

The variety of stopping criteria, associated with the self-labeled methods, makes difficult in estimating the maximum number of iterations performed by each method. For simplicity, we focus the temporal analysis on the complexity related to the execution of each iteration in the main loop of the method. The algorithmic complexity is based on the current number of unlabeled examples ($u$) and labeled examples ($l$) at the beginning of the iteration. Table 1 includes the time analysis of each method. The functions $T_t$ and $T_c$ represent the time cost associated with an specific learning scheme in the task of training the model and classifying new instances, respectively. $l'$ is the number of candidate examples selected to increase the set $L$, and $l''$ is the resulting number of examples after the filtering process. In the case of the SETRED method, the construction of the neighborhood graph has a cubic complexity. For the analysis of the Democratic method, we take into account the existence of $N$ different learning schemes.

## 2.2 Supervised approaches for time series classification

Different approaches have been used to face the time series classification problem such as $k$NN classifiers, decision trees (DT) or support vector machines (SVMs).

The $k$NN classifier has been widely applied in the time series context [28,45]. This classifier approximates the confidence in terms of dissimilarity between instances. There are several distance measures presented in the literature that have been used for evaluating dissimilarity between time series: lock-step measures (Euclidean), feature-based measures (Fourier coefficients), model-based measures (autocorrelation functions), and elastic measures.

The construction of DT is another approach applied to time series classification. Yamada et al. [66] propose two binary split tests called the standard-example and the cluster-example. The former selects an existing instance as the standard time series, and the members of the

child nodes are selected depending on their distances to this selected instance. The later split searches for two standard time series to bisect the set of instances. A similar idea is followed by Balakrishnan and Madigan [4] with a clustering-goodness criterion which searches for the pair of time series that best bisects the set of instances. In both works, the dynamic time warping (DTW [54]) distance is used. A new split criterion based on an adaptive metric that covers both behavior and value proximities is developed in Douzal-Chouakria and Amblard [21].

On the other hand, SVMs are a popular technique that has been applied to time series classification. The work of Pree et al. [47] compares several similarity measures used as kernel function in SVM. In contrast to other learning approaches, the performance of the SVM constructed with Euclidean distance significantly outperforms those obtained using DTW distance. The reason of this behavior has been analyzed in multiple works [37,68]. It is caused by the indefiniteness of the kernel constructed with DTW. The use of classical recursive elastic distances to construct recursive edit distance kernels is addressed in Marteau and Gibet [42]. The kernels constructed in this way are positive definite if some sufficient conditions are satisfied. Moreover, the construction of a weighted DTW kernel to classify time series data is proposed in Jeong and Jayaraman [33].

In this study, we use as base learners three instance-based classifiers representative for those classification approaches, namely *k*NN, DT and SVM.

## 3 Experimental framework

This section presents the information related to the datasets involved in the study in Sect. 3.1. The performance measures and the configuration parameters of the algorithms used are addressed in Sects. 3.2 and 3.3, respectively.

### 3.1 Datasets

The experimentation is based on 35 standard classification datasets taken from public available repositories [15,60]. Table 2 summarizes the main properties of the selected datasets. The datasets involved in this study contain between 56 and 9236 instances, the time series length ranges from 24 to 1882, and the number of classes varies between 2 and 14. For each dataset, the time series are z-normalized, following the recommendation in Rakthanmanon et al. [50].

The datasets are randomly divided using a fivefold cross-validation procedure. Each training partition (4/5 of the total set of examples) is randomly divided into two sets: $L$ and $U$, of labeled and unlabeled (i.e., the labels are withheld and not available to the algorithm) examples, respectively. Following the approach of Triguero et al. [57] and Wang et al. [59], we do not attempt to keep the class proportion in the $L$ and $U$ sets the same as in the whole training set. The class label of the instances selected to form the set $U$ is removed. We make sure that every class has been represented in $L$.

With the purpose of studying the influence of the amount of labeled data, we take different ratios when dividing the training set. In our experiments, three ratios are used: 10, 20 and 30%. For instance, assuming a training partition which contains 100 examples, when the labeled rate is 10%, 10 examples are put into $L$ with their labels while the remaining 90 examples are put into $U$ without their labels. Note that the test partition (25 examples) is kept aside to evaluate the inductive performance of the learned hypothesis.

**Table 2** Summary description of the times series datasets

| Datasets | Number of instances | Time series length | Number of classes |
|---|---|---|---|
| CBF | 930 | 128 | 3 |
| Chlorine | 4307 | 166 | 3 |
| CinC_ECG_t | 1420 | 1639 | 4 |
| Coffee | 56 | 286 | 2 |
| Cricket_X | 780 | 300 | 12 |
| Cricket_Y | 780 | 300 | 12 |
| Cricket_Z | 780 | 300 | 12 |
| ECG | 2026 | 85 | 2 |
| ECGFiveD | 884 | 136 | 2 |
| FaceAll | 2250 | 131 | 14 |
| FacesUCR | 2250 | 131 | 14 |
| Fish | 350 | 463 | 7 |
| Gun_Point | 200 | 150 | 2 |
| Haptics | 463 | 1092 | 5 |
| InlineS | 650 | 1882 | 7 |
| Italy | 1096 | 24 | 2 |
| Lighting2 | 121 | 637 | 2 |
| Lighting7 | 143 | 319 | 7 |
| MALLAT | 2400 | 1024 | 8 |
| MedicalI | 1141 | 99 | 10 |
| MoteStrain | 1272 | 84 | 2 |
| Olive | 60 | 570 | 4 |
| OSULeaf | 442 | 427 | 6 |
| Sony | 621 | 70 | 2 |
| SonyII | 980 | 65 | 2 |
| StarLightC | 9236 | 1024 | 3 |
| Synthetic | 600 | 60 | 6 |
| Trace | 200 | 275 | 4 |
| Two | 5000 | 128 | 4 |
| TwoLeadECG | 1162 | 82 | 2 |
| uWaveGL_X | 4478 | 315 | 8 |
| uWaveGL_Y | 4478 | 315 | 8 |
| uWaveGL_Z | 4477 | 315 | 8 |
| Wafer | 7164 | 152 | 2 |
| Yoga | 3300 | 426 | 2 |

### 3.2 Performance measures

With the aim of measuring the effectiveness of the classification performed by the SSC techniques, we use two classical statistics: accuracy rate [63] and Cohen's kappa rate [8]. The two measures are briefly explained as follows:

**Table 3** Parameter specification for the base learners and self-labeled methods involved in the experimentation

| Algorithm | Parameters |
|---|---|
| *k*NN | Number of neighbors = 1 |
| DT | Minimum number of objects per leaf = 3, impurity level = 0.1, behavior influence = 2.0 |
| SVM | $C = \{2^{-5}, 2^{-4}, \ldots, 1, \ldots, 2^{5}\}$, Kernel type = *RBF*, $\sigma = \{2^{-5}, 2^{-4}, \ldots, 1, \ldots, 2^{5}\}$, cross-validation = threefolds |
| SelfT | Max iterations = $\min\{50, \lceil 0.7 \cdot |U|/\text{InstPerIter} \rceil\}$ |
| SETRED | Max iterations = $\min\{50, \lceil 0.7 \cdot |U|/\text{InstPerIter} \rceil\}$, significance threshold = 0.05 |
| SNNRCE | Significance threshold = 0.05 |
| TriT | No parameters specified |
| Democratic | Classifiers = 1NN, DT and SVM |

– Accuracy: This measure reflects the agreement between the observed and predicted classes. It is a simple metric commonly employed for assessing the performance of classifiers.
– Cohen's kappa: This measure takes into account the successful hits that would be generated simply by chance. Cohen's kappa ranges from −1 to 1, where a value of 0 means there is no agreement, a value of 1 indicates total agreement, and a negative value indicates that the prediction is in the opposite direction.

### 3.3 Algorithms used and parameters

In this section, we specify the configuration parameters of all the methods involved in this study. The selected values are uniformly used in all the datasets, and they were mostly selected taking into account the recommendations offered in previous works. The parameters are not optimized for each specific dataset because the main purpose of this experimental study is to compare the general performance of the self-labeled techniques. The configuration parameters are shown in Table 3.

Some of the self-labeled methods have their own built-in stopping criteria, which we use accordingly for these methods. For classifiers which have not a predefined stopping criterion we define it as follows. For each dataset, the self-labeled process stops when it satisfies the first of the following stopping criteria: (i) 70% of the unlabeled instances in the initial set $U$ have been removed and inserted into $L$ or (ii) the algorithm has reached a maximum number of 50 iterations. Here, *InstPerIter* is the number of instances removed from $U$ for each iteration. The stopping criterion proposed facilitates the exploitation of $U$ and avoids the extreme output caused by adding in the base learner all the unlabeled instances from $U$.

Most of the self-labeled methods include one or more base classifier(s). For those methods that support base classifiers from different approaches (SelfT and TriT), we explore all the possible combinations. In this study, we select as a base classifiers three methods that represent influent approaches of time series classification algorithms: *k*NN, DT and SVM.

1NN is a widely used classifier in the time series classification domain. Multiple studies [28,36,55,65] related to time series similarity measures are based on this classifier.

The method proposed in Douzal-Chouakria and Amblard [21] is selected to construct DT specifically designed to classify time series data. This method obtains competitive results, and its split procedure is flexible enough to cover behavior and value proximities. The cost

function $c_b$ to evaluate the proximity between two series is evaluated as $c_b(r) = 2/(1 + \exp(b \cdot \text{Co}(r))) \cdot c(r)$, where $r$ is a mapping between two series, Co is the behavior-based cost function, $c$ is the values-based cost function, and the parameter $b$ modulates the influence of the behavior in the overall cost. This parameter has been empirically fixed to 2 (Table 3). As Co function, we have used a variant of Pearson correlation involving first-order differences, and as $c$ function, we have explored several time series measures.

The kernel function selected to construct the SVMs is Gaussian radial basis function (RBF), i.e., $K_d(x_i, x_j) = \exp(-d(x_i, x_j)^2/(2\sigma^2))$. Most previous studies [42,47,68] use it in combination with a distance measure $d$ selected from the time series domain. Following the methodology in Marteau and Gibet [42], we normalize the pairwise distance matrix in the training stage to limit the search space of the parameters. Specifically, we use a predefined set of $C$ and sigma values to select the most appropriate value during a cross-validation process. Additionally, other kernels were evaluated [16,56], but result of an unfeasible computational cost in the self-labeled context.

Throughout the experimentation, we evaluate five different measures to compute the dissimilarity between instances: Euclidean, DTW, ERP, ACF and FFT. The Sakoe–Chiba band global constraint [54] is used to improve the performance of the elastic measures. Specifically, we fix the window size to 4 and 9% of the time series length for DTW and ERP, respectively, following the recommendation in Kurbalija et al. [36].

## 4 Results and discussion

This section presents the results obtained in the experiments and a detailed discussion of those. We evaluate the performance of the methods in two different settings: results obtained in transductive learning (Sect. 4.1) and inductive learning (Sect. 4.2), under three different ratios of labeled data. Section 4.3 presents an empirical analysis of the run-times obtained by the self-labeled techniques. Section 4.4 addresses the geometrical characteristics of the time series datasets and its influence in the performance of the techniques studied. A comparison between the supervised and semi-supervised learning paradigm is presented in Sect. 4.5. Finally, the discussion of all results is performed in Sect. 4.6.

We use nonparametric statistical tests to contrast the results obtained, following the methodology proposed in García et al. [27]. Concretely, we use the Aligned Friedman test [32] for multiple comparisons to detect statistically significant differences between the evaluated methods and the post-hoc procedure of Hochberg [31] to characterize those differences. In comparisons with only two algorithms involved, we use the Wilcoxon signed rank test, following the recommendation in Demšar [20].

### 4.1 Transductive results

As stated in Sect. 2.1, the main goal of transductive classification is to predict the class of the unlabeled data used during the training phase. Table 4 presents the average accuracy (*Acc*) results of the self-labeled methods involved in this study over the 35 datasets with 10, 20 and 30% of labeled data. The methods are presented in descending order of the accuracy. For those methods that support different base classifiers, we have explored all possible combinations specifying the classifier in the method's name.

The difference between this ranking and the ranking obtained when we sort using the kappa results is denoted as $\Delta\mathbf{K}$: positive values indicate that the method obtains a better position ranking by kappa, negative values indicate the opposite, and zero means no difference. $\Delta K$

**Table 4** Self-labeled methods ordered by the average accuracy results obtained in the transductive phase

| 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|
| (Euclidean) | Acc | $\Delta K$ | (Euclidean) | Acc | $\Delta K$ | (Euclidean) | Acc | $\Delta K$ |
| SETRED | 0.720 | 0 | SETRED | 0.762 | 0 | SETRED | 0.794 | 0 |
| TriT-1NN | 0.718 | 0 | Democratic | 0.760 | 0 | Democratic | 0.793 | 0 |
| Democratic | 0.715 | 0 | TriT-1NN | 0.759 | 0 | TriT-1NN | 0.790 | 0 |
| SelfT-1NN | 0.708 | 0 | SelfT-1NN | 0.750 | 0 | SNNRCE | 0.781 | −2 |
| SNNRCE | 0.707 | 0 | SNNRCE | 0.749 | 0 | TriT-SVM | 0.777 | +1 |
| TriT-SVM | 0.694 | 0 | TriT-SVM | 0.734 | 0 | SelfT-1NN | 0.776 | +1 |
| TriT-DT | 0.635 | 0 | SelfT-SVM | 0.686 | 0 | SelfT-SVM | 0.721 | 0 |
| SelfT-SVM | 0.618 | 0 | TriT-DT | 0.671 | 0 | TriT-DT | 0.694 | 0 |
| SelfT-DT | 0.598 | 0 | SelfT-DT | 0.639 | 0 | SelfT-DT | 0.664 | 0 |
| 10% | | | 20% | | | 30% | | |
| (DTW) | Acc | $\Delta K$ | (DTW) | Acc | $\Delta K$ | (DTW) | Acc | $\Delta K$ |
| TriT-1NN | 0.771 | 0 | TriT-1NN | 0.815 | 0 | TriT-1NN | 0.840 | 0 |
| SETRED | 0.770 | 0 | SETRED | 0.814 | 0 | SETRED | 0.840 | 0 |
| Democratic | 0.758 | 0 | Democratic | 0.802 | 0 | Democratic | 0.829 | 0 |
| SNNRCE | 0.757 | 0 | SNNRCE | 0.800 | 0 | SNNRCE | 0.827 | 0 |
| SelfT-1NN | 0.745 | 0 | SelfT-1NN | 0.787 | 0 | SelfT-1NN | 0.812 | 0 |
| TriT-SVM | 0.732 | 0 | TriT-SVM | 0.782 | 0 | TriT-SVM | 0.806 | 0 |
| TriT-DT | 0.679 | 0 | TriT-DT | 0.718 | −1 | SelfT-SVM | 0.750 | 0 |
| SelfT-DT | 0.638 | −1 | SelfT-SVM | 0.715 | +1 | TriT-DT | 0.748 | 0 |
| SelfT-SVM | 0.638 | +1 | SelfT-DT | 0.681 | 0 | SelfT-DT | 0.710 | 0 |
| 10% | | | 20% | | | 30% | | |
| (ACF) | Acc | $\Delta K$ | (ACF) | Acc | $\Delta K$ | (ACF) | Acc | $\Delta K$ |
| TriT-1NN | 0.689 | 0 | TriT-1NN | 0.720 | 0 | TriT-1NN | 0.743 | 0 |
| SETRED | 0.685 | 0 | SETRED | 0.715 | 0 | SETRED | 0.737 | 0 |
| SNNRCE | 0.681 | 0 | SNNRCE | 0.710 | 0 | SNNRCE | 0.731 | 0 |
| Democratic | 0.677 | 0 | Democratic | 0.709 | 0 | Democratic | 0.730 | 0 |
| SelfT-1NN | 0.655 | 0 | SelfT-1NN | 0.687 | 0 | TriT-DT | 0.708 | −1 |
| TriT-DT | 0.654 | 0 | TriT-DT | 0.684 | 0 | SelfT-1NN | 0.705 | +1 |
| TriT-SVM | 0.640 | 0 | TriT-SVM | 0.683 | 0 | TriT-SVM | 0.699 | −1 |
| SelfT-DT | 0.627 | 0 | SelfT-DT | 0.665 | 0 | SelfT-DT | 0.693 | +1 |
| SelfT-SVM | 0.569 | 0 | SelfT-SVM | 0.632 | 0 | SelfT-SVM | 0.659 | 0 |
| 10% | | | 20% | | | 30% | | |
| (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ |
| SETRED | 0.721 | 0 | SETRED | 0.762 | 0 | SETRED | 0.794 | −1 |
| Democratic | 0.715 | 0 | TriT-1NN | 0.760 | 0 | Democratic | 0.794 | +1 |
| TriT-1NN | 0.715 | 0 | Democratic | 0.759 | 0 | TriT-1NN | 0.790 | 0 |
| SelfT-1NN | 0.709 | 0 | SNNRCE | 0.750 | −1 | SNNRCE | 0.782 | −1 |
| SNNRCE | 0.708 | 0 | SelfT-1NN | 0.749 | +1 | SelfT-1NN | 0.776 | +1 |

**Table 4** continued

| 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|
| (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ |
| TriT-SVM | 0.694 | 0 | TriT-SVM | 0.741 | 0 | TriT-SVM | 0.768 | 0 |
| TriT-DT | 0.636 | 0 | SelfT-SVM | 0.681 | 0 | SelfT-SVM | 0.723 | 0 |
| SelfT-SVM | 0.622 | 0 | TriT-DT | 0.673 | 0 | TriT-DT | 0.696 | 0 |
| SelfT-DT | 0.597 | 0 | SelfT-DT | 0.639 | 0 | SelfT-DT | 0.663 | 0 |
| 10% | | | 20% | | | 30% | | |
| (ERP) | Acc | $\Delta K$ | (ERP) | Acc | $\Delta K$ | (ERP) | Acc | $\Delta K$ |
| SETRED | 0.783 | 0 | Democratic | 0.825 | 0 | Democratic | 0.847 | 0 |
| Democratic | 0.780 | 0 | TriT-1NN | 0.821 | 0 | TriT-1NN | 0.846 | 0 |
| TriT-1NN | 0.779 | 0 | SETRED | 0.820 | 0 | SETRED | 0.845 | 0 |
| TriT-SVM | 0.772 | 0 | SNNRCE | 0.808 | 0 | TriT-SVM | 0.836 | 0 |
| SNNRCE | 0.767 | −1 | TriT-SVM | 0.805 | −1 | SNNRCE | 0.832 | 0 |
| SelfT-1NN | 0.760 | +1 | SelfT-1NN | 0.800 | +1 | SelfT-1NN | 0.823 | 0 |
| SelfT-SVM | 0.724 | 0 | TriT-DT | 0.736 | 0 | TriT-DT | 0.762 | −1 |
| TriT-DT | 0.696 | 0 | SelfT-SVM | 0.722 | 0 | SelfT-SVM | 0.756 | +1 |
| SelfT-DT | 0.689 | 0 | SelfT-DT | 0.697 | 0 | SelfT-DT | 0.722 | 0 |

shows whether or not a certain method benefits from random hits. Table 4 reveals no significant difference between the two orders as only a handful of methods exhibit $\Delta K$ values different from zero.

Figure 1 shows box and whisker plots of the methods under the dissimilarity measures studied. This illustration allows us to visualize in more detail the performance of the self-labeled methods. It shows the gain of accuracy caused by the use of DTW and ERP in comparison with the other measures. The superiority of DTW over Euclidean distance has been addressed in previous studies about time series classification problem. For instance, the extensive study performed by Serrà and Arcos [55] supports this conclusion. Furthermore, the study of Wang et al. [58] reveals that DTW and ERP are clearly superior to Euclidean distance. In this sense, our results confirm the advantage of these elastic measures in the semi-supervised context.

On the other hand, the methods combined with 1NN as a base classifier exhibit the better performance. By contrast, the lowest results are obtained in combination with DT. Moreover, the use of SVM as a base classifier causes a spread behavior of the accuracy values. Figure 2 presents the average results in a bar plot aiming to compare the accuracy values across different labeled ratios.

We perform a comparison of the accuracy among all single learning methods grouped by their learning scheme. This comparison allows us to determine the most successful self-labeled methods for each base classifier.

The Aligned Friedman test, applied to accuracy for all methods that use 1NN as a base classifier, detects significant differences in a significance level of $\alpha = 0.05$ for all comparisons performed. Table 5 shows the rankings obtained. The most accurate method is chosen as the control method for the application of the post-hoc procedure. For Euclidean and FFT distance, the method selected is SETRED in all labeled ratios. For DTW, ACF and ERP, the method

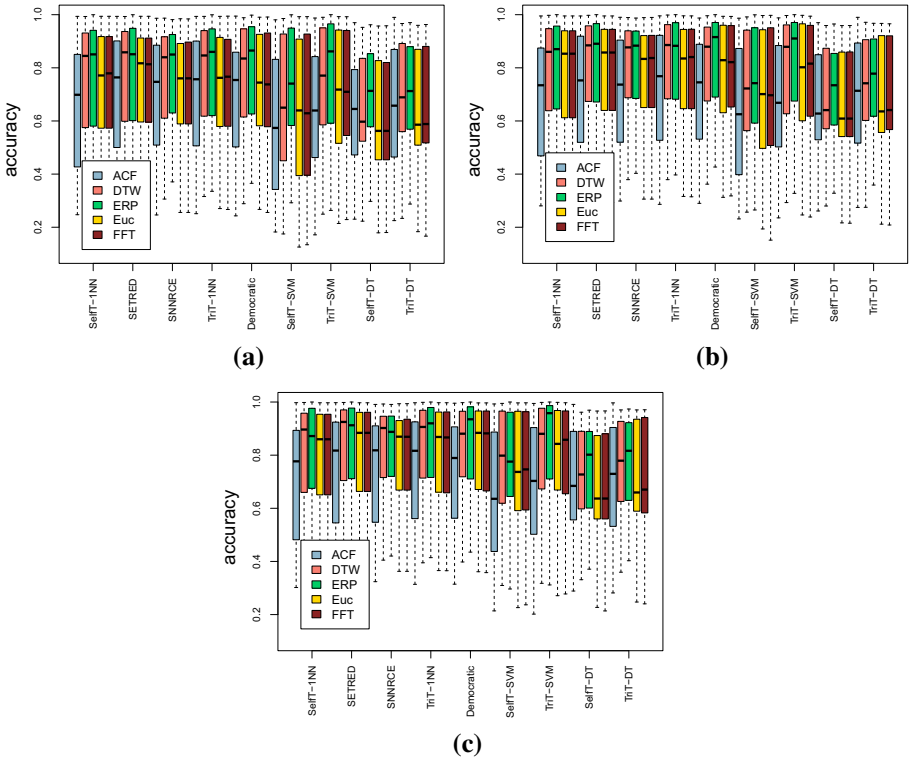



**Fig. 1** *Box* and *whisker plots* for the accuracy in transductive phase. The *bottom* and *top of a box* are the first and third quartiles. The *band inside the box* is the median. **a** 10% labeled data. **b** 20% labeled data. **c** 30% labeled data

selected is TriT in most of the comparisons. SNNRCE and SelfT show the lowest values of accuracy. In the majority of comparisons, these methods are significantly outperformed by the control method, following the Hochberg post-hoc procedure.

Table 6 shows the application of the Wilcoxon signed ranks test to contrast the accuracy of the methods that use DT and SVM as a base classifiers. For all dissimilarity measures and labeled ratios, TriT outperforms SelfT using both base classifiers. The difference obtained results significant on a significance level of $\alpha = 0.05$, with the exception of ERP at 10% of labeled data.

Finally, Table 7 offers a comparison between the most competitive methods from single learning and the multi-learning approach (Democratic). We consider as "competitive" those methods that have not been significantly outperformed more than twice in the 15 comparisons performed (five distance measures $\times$ three labeled ratios) in Tables 5 and 6. Following this criterion, the outstanding methods selected are: SETRED and TriT.

The Aligned Friedman test, applied to accuracy, detects significant differences on a significance level of $\alpha = 0.05$ for all comparisons performed. For the dissimilarity measures DTW and ACF, the control method selected is TriT-1NN in most cases. For the dissimilarity measures Euclidean, FFT and ERP, Democratic is selected as control method in most of the comparisons. SETRED exhibits a competitive behavior because is not outperformed by the control method in any of comparisons. By contrast, TriT-SVM and TriT-DT are signifi-

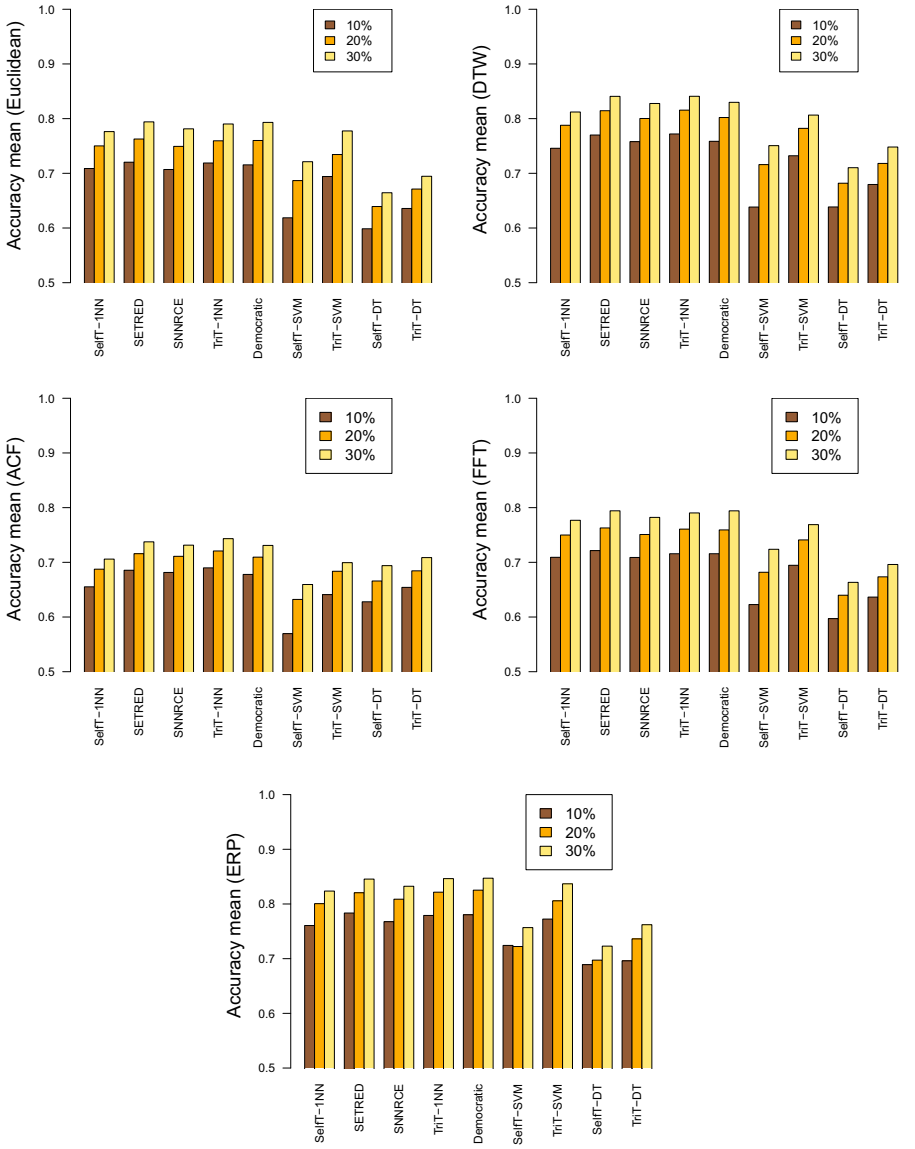



**Fig. 2** *Bar plot* of the comparison between the average accuracy obtained during the transductive phase

cantly outperformed by the control method in most of the comparisons performed, with the exception of ERP where TriT-SVM exhibits a better behavior.

## 4.2 Inductive results

The main target of inductive learning is to classify instances not included in the training phase. This analysis is useful to test the previous learned hypotheses and their generalization abilities. Table 8 shows the average obtained using all dissimilarity measures studied. Figure 4

**Table 5** Aligned Friedman ranking of the accuracy using 1NN as a base classifier

| *1NN* | Algorithm | 10% | | 20% | | 30% | |
|---|---|---|---|---|---|---|---|
| | | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ |
| (Euc) | SETRED | **55.2** | – | **53.7** | – | **49.9** | – |
| | TriT | 60.8 | 0.56 | 61.3 | 0.43 | 60.9 | 0.25 |
| | SelfT | 77.2 | **0.04** | 77.2 | **0.03** | 86.0 | $\mathbf{5.8} \times 10^{-4}$ |
| | SNNRCE | 88.7 | $\mathbf{1.6} \times 10^{-3}$ | 89.6 | $\mathbf{6.4} \times 10^{-5}$ | 85.0 | $\mathbf{5.8} \times 10^{-4}$ |
| (DTW) | TriT | **50.0** | – | **47.9** | – | 50.4 | 0.96 |
| | SETRED | 57.0 | 0.47 | 50.9 | 0.75 | **49.9** | – |
| | SNNRCE | 83.2 | $\mathbf{1.2} \times 10^{-3}$ | 89.0 | $\mathbf{4.5} \times 10^{-5}$ | 81.5 | $\mathbf{2.2} \times 10^{-3}$ |
| | SelfT | 91.7 | $\mathbf{5.1} \times 10^{-5}$ | 94.0 | $\mathbf{6.0} \times 10^{-6}$ | 100.0 | $\mathbf{1.0} \times 10^{-6}$ |
| (ACF) | TriT | **50.2** | – | **45.7** | – | **44.5** | – |
| | SETRED | 63.9 | 0.15 | 59.8 | 0.14 | 62.9 | 0.06 |
| | SNNRCE | 68.5 | 0.11 | 70.5 | **0.02** | 72.3 | $\mathbf{8.2} \times 10^{-3}$ |
| | SelfT | 99.2 | $\mathbf{1.0} \times 10^{-6}$ | 105.9 | **0.0** | 102.2 | **0.0** |
| (FFT) | SETRED | **55.0** | – | **55.3** | – | **49.8** | – |
| | TriT | 66.4 | 0.23 | 60.0 | 0.62 | 60.3 | 0.27 |
| | SNNRCE | 84.4 | $\mathbf{7.3} \times 10^{-3}$ | 88.7 | $\mathbf{1.7} \times \times 10^{-3}$ | 85.7 | $\mathbf{4.2} \times 10^{-4}$ |
| | SelfT | 76.0 | 0.06 | 77.9 | **0.03** | 86.1 | $\mathbf{4.2} \times 10^{-4}$ |
| (ERP) | TriT | 55.5 | 0.68 | **50.6** | – | **50.7** | – |
| | SETRED | **51.5** | – | 56.7 | 0.52 | 57.5 | 0.48 |
| | SelfT | 89.7 | $\mathbf{2.5} \times 10^{-4}$ | 92.0 | $\mathbf{5.8} \times 10^{-5}$ | 91.9 | $\mathbf{6.4} \times 10^{-5}$ |
| | SNNRCE | 85.1 | $\mathbf{1.0} \times 10^{-3}$ | 82.6 | $\mathbf{1.8} \times 10^{-3}$ | 81.7 | $\mathbf{2.8} \times 10^{-3}$ |

Adjusted $p$ values for the post-hoc procedure of Hochberg

**Table 6** Wilcoxon signed ranks test of the accuracy for DT and SVM as a base classifiers

| | Algorithms | 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Neg | Pos | $p_{\text{value}}$ | Neg | Pos | $p_{\text{value}}$ | Neg | Pos | $p_{\text{value}}$ |
| *DT* | | | | | | | | | | |
| (Euc) | TriT-SelfT | 4 | **31** | $\mathbf{4.0} \times 10^{-6}$ | 4 | **31** | $\mathbf{1.0} \times 10^{-5}$ | 3 | **32** | $\mathbf{1.0} \times 10^{-6}$ |
| (DTW) | TriT-SelfT | 3 | **32** | $\mathbf{1.0} \times 10^{-6}$ | 5 | **30** | $\mathbf{2.0} \times 10^{-6}$ | 1 | **34** | **0.0** |
| (ACF) | TriT-SelfT | 9 | **26** | $\mathbf{8.0} \times 10^{-3}$ | 10 | **25** | $\mathbf{7.0} \times 10^{-3}$ | 8 | **27** | $\mathbf{6.0} \times 10^{-3}$ |
| (FFT) | TriT-SelfT | 5 | **30** | $\mathbf{2.0} \times 10^{-6}$ | 3 | **32** | $\mathbf{2.0} \times 10^{-6}$ | 3 | **32** | $\mathbf{1.0} \times 10^{-6}$ |
| (ERP) | TriT-SelfT | 16 | **19** | 0.53 | 2 | **33** | $\mathbf{1.0} \times 10^{-6}$ | 2 | **33** | **0.0** |
| *SVM* | | | | | | | | | | |
| (Euc) | TriT-SelfT | 5 | **30** | $\mathbf{7.0} \times 10^{-6}$ | 12 | **23** | $\mathbf{4.0} \times 10^{-3}$ | 8 | **27** | $\mathbf{1.0} \times 10^{-4}$ |
| (DTW) | TriT-SelfT | 5 | **30** | $\mathbf{3.0} \times 10^{-6}$ | 8 | **27** | $\mathbf{3.2} \times 10^{-5}$ | 7 | **28** | $\mathbf{2.1} \times 10^{-5}$ |
| (ACF) | TriT-SelfT | 9 | **26** | $\mathbf{2.7} \times 10^{-4}$ | 6 | **29** | $\mathbf{2.2} \times 10^{-4}$ | 12 | **23** | $\mathbf{3.0} \times 10^{-3}$ |
| (FFT) | TriT-SelfT | 7 | **28** | $\mathbf{5.0} \times 10^{-6}$ | 8 | **27** | $\mathbf{3.1} \times 10^{-4}$ | 10 | **25** | $\mathbf{2.8} \times 10^{-3}$ |
| (ERP) | TriT-SelfT | 16 | **19** | 0.09 | 5 | **30** | $\mathbf{2.1} \times 10^{-5}$ | 7 | **26** | $\mathbf{1.4} \times 10^{-4}$ |

The number of negative and positive ranks is shown in conjunction with the $p$ value

**Table 7** Aligned Friedman ranking of the accuracy of the most competent methods for each learning approach using the dissimilarity measures studied

| Distance | Algorithm | 10% | | 20% | | 30% | |
|---|---|---|---|---|---|---|---|
| | | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ |
| (Euc) | TriT-1NN | **66.4** | – | 69.8 | 0.47 | 72.6 | 0.35 |
| | Democratic | 67.3 | 0.93 | **61.0** | – | **61.3** | – |
| | SETRED | 71.6 | 0.93 | 72.0 | 0.47 | 73.5 | 0.35 |
| | TriT-SVM | 92.7 | 0.08 | 91.9 | **0.03** | 85.2 | 0.14 |
| | TriT-DT | 141.8 | **0.0** | 145.1 | **0.0** | 147.3 | **0.0** |
| (DTW) | TriT-1NN | **56.9** | – | **58.4** | – | **57.9** | – |
| | SETRED | 64.9 | 0.51 | 61.6 | 0.79 | 63.8 | 0.62 |
| | Democratic | 71.7 | 0.44 | 72.9 | 0.46 | 71.4 | 0.52 |
| | TriT-SVM | 105.0 | $\mathbf{2.1} \times 10^{-4}$ | 100.6 | $\mathbf{1.5} \times 10^{-3}$ | 101.1 | $\mathbf{1.1} \times 10^{-3}$ |
| | TriT-DT | 141.3 | **0.0** | 146.2 | **0.0** | 145.6 | **0.0** |
| (ACF) | TriT-1NN | **55.1** | – | **58.2** | – | **56.7** | – |
| | SETRED | 62.5 | 0.54 | 67.8 | 0.42 | 71.2 | 0.23 |
| | Democratic | 71.5 | 0.35 | 75.8 | 0.28 | 73.6 | 0.23 |
| | TriT-SVM | 132.2 | **0.0** | 113.7 | $\mathbf{1.4} \times 10^{-5}$ | 121.8 | **0.0** |
| | TriT-DT | 118.5 | **0.0** | 124.4 | **0.0** | 116.5 | $\mathbf{2.0} \times 10^{-6}$ |
| (FFT) | Democratic | **63.2** | – | **62.6** | – | **60.2** | – |
| | TriT-1NN | 70.0 | 0.57 | 68.9 | 0.59 | 70.4 | 0.40 |
| | SETRED | 70.3 | 0.57 | 73.9 | 0.59 | 73.6 | 0.40 |
| | TriT-SVM | 93.8 | **0.03** | 89.6 | **0.07** | 90.4 | **0.03** |
| | TriT-DT | 142.4 | **0.0** | 144.8 | **0.0** | 145.2 | **0.0** |
| (ERP) | Democratic | **69.0** | – | **57.8** | – | **66.3** | – |
| | TriT-1NN | 75.0 | 0.75 | 72.3 | 0.23 | 71.4 | 0.67 |
| | SETRED | 72.8 | 0.75 | 74.6 | 0.23 | 75.3 | 0.67 |
| | TriT-SVM | 73.8 | 0.75 | 86.4 | 0.05 | 76.3 | 0.67 |
| | TriT-DT | 149.2 | **0.0** | 148.7 | **0.0** | 150.4 | **0.0** |

Adjusted $p$ values for the post-hoc procedure of Hochberg

shows box and whisker plots of the same results grouped by ratios. Figure 3 shows a bar plot of the average accuracy reflecting the improvement obtained by increasing the amount of labeled examples.

Once more, we perform a comparison of the accuracy among all single learning methods grouped by their learning scheme. The aligned Friedman test, applied to the accuracy of all methods that use 1NN as base learning scheme, detects significant differences for all comparisons performed. Table 9 shows the rankings obtained. The control method selected is SETRED in most of the comparisons except for ACF where TriT is selected in two labeled ratios. In general, SNNRCE and SelfT show the lowest values of accuracy and are significantly outperformed by the control method in most of the comparisons following the Hochberg post-hoc procedure.

Table 10 shows the application of the Wilcoxon signed ranks test to the accuracy for the methods that use DT and SVM as a base classifiers. For all dissimilarity measures and labeled

**Table 8** Self-labeled methods ordered by the average accuracy results obtained in the inductive phase

| 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|
| (Euclidean) | Acc | $\Delta K$ | (Euclidean) | Acc | $\Delta K$ | (Euclidean) | Acc | $\Delta K$ |
| Democratic | 0.719 | 0 | SETRED | 0.763 | 0 | SETRED | 0.794 | −1 |
| SETRED | 0.718 | 0 | Democratic | 0.757 | 0 | Democratic | 0.794 | +1 |
| TriT-1NN | 0.715 | 0 | TriT-1NN | 0.756 | 0 | TriT-1NN | 0.788 | 0 |
| SelfT-1NN | 0.711 | 0 | SNNRCE | 0.754 | −1 | SNNRCE | 0.788 | 0 |
| SNNRCE | 0.707 | 0 | SelfT-1NN | 0.753 | +1 | SelfT-1NN | 0.784 | 0 |
| TriT-SVM | 0.693 | 0 | TriT-SVM | 0.734 | 0 | TriT-SVM | 0.778 | 0 |
| TriT-DT | 0.633 | −1 | SelfT-SVM | 0.694 | 0 | SelfT-SVM | 0.739 | 0 |
| SelfT-SVM | 0.627 | +1 | TriT-DT | 0.667 | 0 | TriT-DT | 0.693 | 0 |
| SelfT-DT | 0.602 | 0 | SelfT-DT | 0.641 | 0 | SelfT-DT | 0.665 | 0 |
| 10% | | | 20% | | | 30% | | |
| (DTW) | Acc | $\Delta K$ | (DTW) | Acc | $\Delta K$ | (DTW) | Acc | $\Delta K$ |
| SETRED | 0.771 | 0 | SETRED | 0.813 | 0 | SETRED | 0.843 | 0 |
| TriT-1NN | 0.768 | 0 | TriT-1NN | 0.810 | 0 | TriT-1NN | 0.839 | 0 |
| SNNRCE | 0.764 | 0 | SNNRCE | 0.807 | 0 | SNNRCE | 0.838 | 0 |
| Democratic | 0.760 | 0 | Democratic | 0.801 | 0 | Democratic | 0.831 | 0 |
| SelfT-1NN | 0.748 | 0 | SelfT-1NN | 0.791 | 0 | SelfT-1NN | 0.824 | 0 |
| TriT-SVM | 0.732 | 0 | TriT-SVM | 0.783 | 0 | TriT-SVM | 0.806 | 0 |
| TriT-DT | 0.677 | 0 | SelfT-SVM | 0.725 | 0 | SelfT-SVM | 0.762 | 0 |
| SelfT-SVM | 0.643 | 0 | TriT-DT | 0.712 | 0 | TriT-DT | 0.743 | 0 |
| SelfT-DT | 0.636 | 0 | SelfT-DT | 0.679 | 0 | SelfT-DT | 0.710 | 0 |
| 10% | | | 20% | | | 30% | | |
| (ACF) | Acc | $\Delta K$ | (ACF) | Acc | $\Delta K$ | (ACF) | Acc | $\Delta K$ |
| TriT-1NN | 0.689 | 0 | TriT-1NN | 0.720 | 0 | SETRED | 0.744 | 0 |
| SNNRCE | 0.688 | −1 | SETRED | 0.717 | 0 | TriT-1NN | 0.741 | 0 |
| SETRED | 0.685 | +1 | SNNRCE | 0.714 | 0 | SNNRCE | 0.739 | 0 |
| Democratic | 0.674 | 0 | Democratic | 0.708 | 0 | Democratic | 0.732 | 0 |
| SelfT-1NN | 0.659 | 0 | SelfT-1NN | 0.696 | 0 | SelfT-1NN | 0.717 | 0 |
| TriT-DT | 0.654 | 0 | TriT-SVM | 0.687 | −1 | TriT-DT | 0.711 | 0 |
| TriT-SVM | 0.643 | 0 | TriT-DT | 0.684 | +1 | TriT-SVM | 0.702 | −1 |
| SelfT-DT | 0.629 | 0 | SelfT-DT | 0.664 | 0 | SelfT-DT | 0.697 | +1 |
| SelfT-SVM | 0.565 | 0 | SelfT-SVM | 0.632 | 0 | SelfT-SVM | 0.667 | 0 |
| 10% | | | 20% | | | 30% | | |
| (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ |
| SETRED | 0.720 | 0 | SETRED | 0.764 | 0 | SETRED | 0.794 | 0 |
| Democratic | 0.716 | 0 | Democratic | 0.758 | 0 | Democratic | 0.791 | 0 |
| TriT-1NN | 0.713 | −1 | TriT-1NN | 0.757 | 0 | SNNRCE | 0.789 | -1 |
| SelfT-1NN | 0.711 | +1 | SNNRCE | 0.755 | −1 | TriT-1NN | 0.788 | +1 |
| SNNRCE | 0.710 | 0 | SelfT-1NN | 0.753 | +1 | SelfT-1NN | 0.785 | 0 |

**Table 8** continued

| 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|
| (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ | (FFT) | Acc | $\Delta K$ |
| TriT-SVM | 0.697 | 0 | TriT-SVM | 0.742 | 0 | TriT-SVM | 0.771 | 0 |
| TriT-DT | 0.638 | −1 | SelfT-SVM | 0.687 | 0 | SelfT-SVM | 0.739 | 0 |
| SelfT-SVM | 0.628 | +1 | TriT-DT | 0.673 | 0 | TriT-DT | 0.691 | 0 |
| SelfT-DT | 0.601 | 0 | SelfT-DT | 0.642 | 0 | SelfT-DT | 0.663 | 0 |
| 10% | | | 20% | | | 30% | | |
| (ERP) | Acc | $\Delta K$ | (ERP) | Acc | $\Delta K$ | (ERP) | Acc | $\Delta K$ |
| SETRED | 0.779 | 0 | SETRED | 0.825 | 0 | SETRED | 0.845 | 0 |
| Democratic | 0.778 | 0 | Democratic | 0.822 | 0 | TriT-1NN | 0.844 | 0 |
| TriT-1NN | 0.774 | 0 | TriT-1NN | 0.820 | 0 | Democratic | 0.842 | 0 |
| TriT-SVM | 0.772 | 0 | SNNRCE | 0.815 | 0 | SNNRCE | 0.841 | 0 |
| SNNRCE | 0.768 | −1 | TriT-SVM | 0.808 | −1 | TriT-SVM | 0.836 | 0 |
| SelfT-1NN | 0.763 | +1 | SelfT-1NN | 0.808 | +1 | SelfT-1NN | 0.832 | 0 |
| SelfT-SVM | 0.730 | 0 | TriT-DT | 0.732 | −1 | SelfT-SVM | 0.772 | 0 |
| TriT-DT | 0.694 | 0 | SelfT-SVM | 0.730 | +1 | TriT-DT | 0.753 | 0 |
| SelfT-DT | 0.684 | 0 | SelfT-DT | 0.693 | 0 | SelfT-DT | 0.726 | 0 |

ratios, TriT outperforms significantly SelfT using both base classifiers, with the exception of SVM under ERP at 10% of labeled data.

Finally, Table 11 offers a comparison between the most competitive methods from single learning and the multi-learning approach. Once more, the outstanding methods selected are: SETRED and TriT. The aligned Friedman test, applied to accuracy, detects significant differences for all comparisons performed. For the dissimilarity measures Euclidean, FFT and ERP, the control method selected is Democratic in all comparisons. For ACF, TriT-1NN is selected as control method in most of the comparisons. SETRED exhibits the best behavior under the dissimilarity measure DTW. TriT-SVM and TriT-DT are significantly outperformed in most of the comparisons by the control method, with the exception of FFT and ERP where TriT-SVM exhibits a competitive behavior.

### 4.3 Experimental run-times

From the temporal analysis performed in Sect. 2.1, it is clear that the main source of temporal complexity is related to the successive operations of training the model(s) and classifying instances. The cost associated with this operations directly depends on the learning scheme(s) used. In this section, we present an empirical analysis of the run-times based on a sample of 20 datasets included in the experimentation. All experiments were performed in a cluster conformed by 46 nodes, each one equipped with an Intel® Core™ i7-930 processor at 2.8 GHz and 24 GB of RAM memory. Under GNU/Linux, we ran the experiments using R version 3.3.1 [48].

During the training and classification process, we provide the time series datasets to the self-labeled techniques using a distance matrix form. This avoids the repetitions of distance calculations, and therefore it reduces the running time. For that reason, the time consumption associated with the distance matrices computation are not considered in the current analysis.

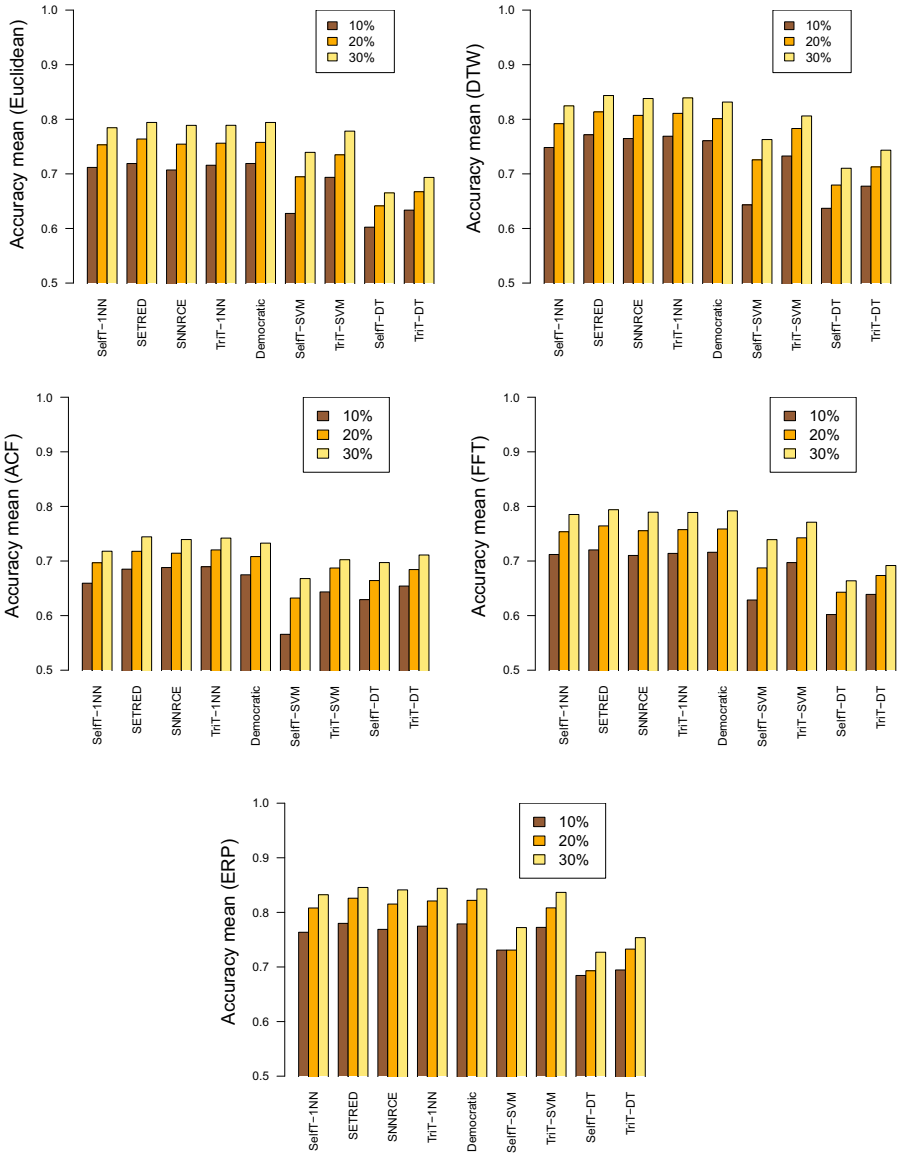



**Fig. 3** *Bar plot* of the comparison between the average accuracy obtained during the transductive phase

Table 12 contains the run-times of the self-labeled techniques using a sequential execution of the fivefold cross-validation procedure. TriT-1NN obtains the lower run-times in all the cases. The 1-NN base classifier produces the shortest run-times, whereas the SVM produces the longest ones. This is caused by the time consumption involved in the construction of SVM with the threefold cross-validation process to adjust the parameters $C$ and $\sigma$. Democratic obtains expensive run-time because it trains a classifier for each learning scheme. SETRED and SNNRCE obtain competitive results as the base classifier trained is 1NN.

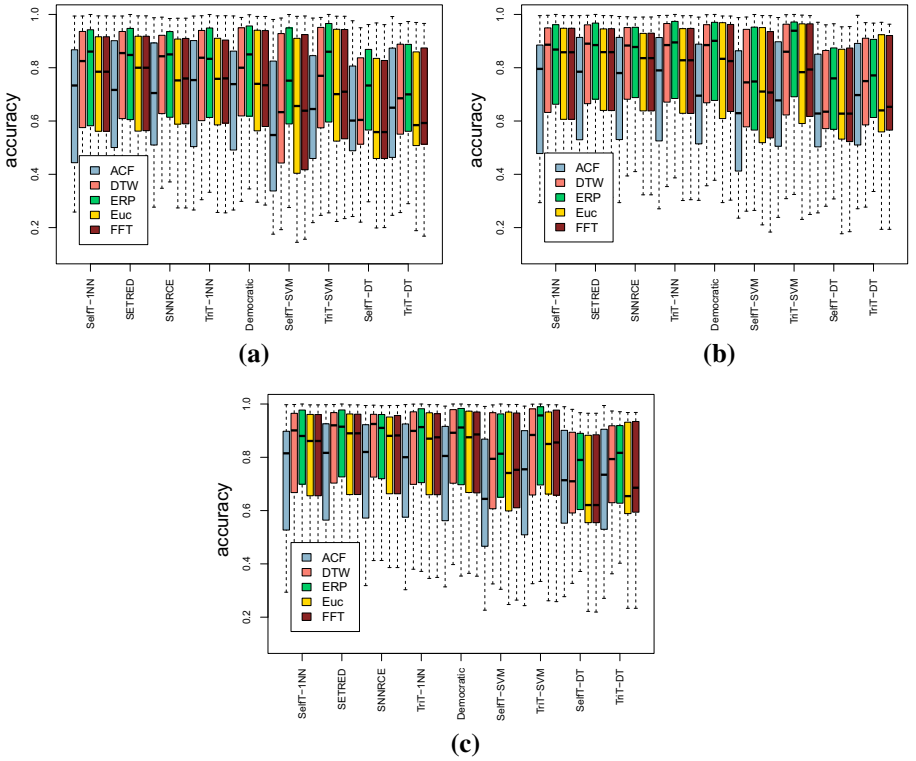



**Fig. 4** *Box* and *whisker plots* for the accuracy in inductive phase. **a** 10% labeled data. **b** 20% labeled data. **c** 30% labeled data

### 4.4 Impact of the geometrical characteristics of datasets

The geometrical characteristics of the datasets affect the performance of the self-labeled methods. The overlapping of samples is a common source of difficulty in the classification process. In addition, the reduced labeled sample in the SSC framework and the high dimensionality of time series data introduces another layer of complexity. In Wang et al. [59], the overlapping of samples from different classes is investigated in order to offer an explanation of the decreasing performance experimented by the SSC algorithms in datasets that suffer this problem.

We follow a similar idea by computing the neighbors of each training instance (from $U$) in a neighborhood graph constructed from $L \cup U$. The proportion of neighbors (from $L$) with a different class with respect to the training instance analyzed, is used as an overlapping measure. Table 13 includes this overlapping measured averaged for all training examples of each dataset. The neighborhood graph was computed for each dissimilarity measure and proportion of labeled data. The datasets with high proportions of neighbors with different class are related to high levels of overlap between classes.

In order to investigate the impact of the overlapping in the classification performance, we correlate the values of Table 13 with the accuracy obtained with the self-labeled techniques. Figure 5 shows the correlations obtained between the two variables studied (accuracy and overlapping). For all dissimilarity measures and proportions of labeled data studied,

**Table 9** Aligned Friedman ranking of the accuracy using 1NN as a base classifier

| *1NN* | Algorithm | 10% | | 20% | | 30% | |
|---|---|---|---|---|---|---|---|
| | | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ |
| (Euc) | SETRED | **56.7** | – | **53.5** | – | **53.3** | – |
| | TriT | 67.2 | 0.27 | 72.5 | 0.06 | 68.4 | **0.04** |
| | SelfT | 69.4 | 0.27 | 76.2 | **0.04** | 86.2 | **0.03** |
| | SNNRCE | 88.5 | $\mathbf{3.0 \times 10^{-3}}$ | 79.7 | **0.04** | 74.0 | **0.02** |
| (DTW) | TriT | **55.7** | – | 63.1 | 0.44 | 63.1 | 0.29 |
| | SETRED | 58.5 | 0.77 | **55.7** | – | **52.9** | – |
| | SNNRCE | 72.8 | 0.15 | 73.3 | 0.14 | 73.5 | 0.06 |
| | SelfT | 94.7 | $\mathbf{1.7 \times 10^{-4}}$ | 89.8 | $\mathbf{1.3 \times 10^{-3}}$ | 92.4 | $\mathbf{1.4 \times 10^{-4}}$ |
| (ACF) | TriT | **52.6** | – | **51.1** | – | 58.8 | 0.40 |
| | SETRED | 68.5 | 0.20 | 61.9 | 0.26 | **50.7** | – |
| | SNNRCE | 59.8 | 0.45 | 67.9 | 0.16 | 70.0 | 0.09 |
| | SelfT | 100.9 | $\mathbf{2.0 \times 10^{-6}}$ | 101.0 | $\mathbf{1.0 \times 10^{-6}}$ | 102.3 | **0.0** |
| (FFT) | SETRED | **55.4** | – | **53.5** | – | **53.9** | – |
| | TriT | 72.6 | 0.09 | 73.9 | **0.03** | 68.9 | 0.12 |
| | SNNRCE | 82.3 | **0.01** | 78.3 | **0.03** | 73.3 | 0.09 |
| | SelfT | 71.5 | 0.09 | **0.03** | $\mathbf{2.0 \times 10^{-6}}$ | 85.8 | $\mathbf{3.0 \times 10^{-3}}$ |
| (ERP) | SETRED | **55.5** | – | **51.1** | – | **61.0** | – |
| | TriT | 60.6 | 0.59 | 64.5 | 0.16 | 62.1 | 0.90 |
| | SelfT | 87.7 | $\mathbf{2.7 \times 10^{-3}}$ | 89.6 | $\mathbf{2.1 \times 10^{-4}}$ | 87.1 | **0.02** |
| | SNNRCE | 78.1 | **0.03** | 76.6 | **0.01** | 71.7 | 0.53 |

Adjusted $p$ values for the post-hoc procedure of Hochberg

**Table 10** Wilcoxon signed ranks test of the accuracy for DT and SVM as a base classifiers

| | Algorithms | 10% | | | 20% | | | 30% | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Neg | Pos | $p_{\text{value}}$ | Neg | Pos | $p_{\text{value}}$ | Neg | Pos | $p_{\text{value}}$ |
| *DT* | | | | | | | | | | |
| (Euc) | TriT-SelfT | 6 | **28** | $\mathbf{1.3 \times 10^{-5}}$ | 6 | **29** | $\mathbf{7.1 \times 10^{-5}}$ | 3 | **32** | $\mathbf{5.0 \times 10^{-6}}$ |
| (DTW) | TriT-SelfT | 3 | **32** | $\mathbf{2.0 \times 10^{-6}}$ | 6 | **29** | $\mathbf{1.0 \times 10^{-5}}$ | 3 | **31** | $\mathbf{5.3 \times 10^{-5}}$ |
| (ACF) | TriT-SelfT | 7 | **27** | $\mathbf{6.7 \times 10^{-3}}$ | 9 | **26** | $\mathbf{6.8 \times 10^{-3}}$ | 9 | **26** | **0.01** |
| (FFT) | TriT-SelfT | 4 | **30** | $\mathbf{1.1 \times 10^{-5}}$ | 4 | **31** | $\mathbf{3.0 \times 10^{-6}}$ | 5 | **30** | $\mathbf{6.0 \times 10^{-6}}$ |
| (ERP) | TriT-SelfT | 10 | **24** | **0.04** | 4 | **30** | $\mathbf{1.0 \times 10^{-6}}$ | 7 | **28** | $\mathbf{2.0 \times 10^{-4}}$ |
| *SVM* | | | | | | | | | | |
| (Euc) | TriT-SelfT | 6 | **29** | $\mathbf{2.6 \times 10^{-5}}$ | 12 | **23** | $\mathbf{8.0 \times 10^{-3}}$ | 9 | **25** | $\mathbf{1.8 \times 10^{-3}}$ |
| (DTW) | TriT-SelfT | 3 | **32** | $\mathbf{4.0 \times 10^{-6}}$ | 7 | **27** | $\mathbf{6.1 \times 10^{-5}}$ | 9 | **26** | $\mathbf{2.2 \times 10^{-4}}$ |
| (ACF) | TriT-SelfT | 6 | **28** | $\mathbf{1.8 \times 10^{-5}}$ | 7 | **28** | $\mathbf{2.8 \times 10^{-5}}$ | 10 | **25** | $\mathbf{3.7 \times 10^{-3}}$ |
| (FFT) | TriT-SelfT | 3 | **32** | $\mathbf{1.0 \times 10^{-6}}$ | 8 | **27** | $\mathbf{4.0 \times 10^{-4}}$ | 11 | **23** | **0.01** |
| (ERP) | TriT-SelfT | 15 | **20** | 0.12 | 4 | **31** | $\mathbf{6.0 \times 10^{-6}}$ | 6 | **27** | $\mathbf{1.0 \times 10^{-4}}$ |

The number of negative and positive ranks is shown in conjunction with the $p$ value

**Table 11** Aligned Friedman ranking of the accuracy of the competent methods for each learning approach using the dissimilarity measures studied

| Distance | Algorithm | 10% | | 20% | | 30% | |
|---|---|---|---|---|---|---|---|
| | | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ | Rank | $p_{\text{Hochberg}}$ |
| (Euc) | Democratic | **61.0** | – | **63.9** | – | **63.2** | – |
| | SETRED | 74.0 | 0.40 | 66.5 | 0.82 | 71.0 | 0.52 |
| | TriT-1NN | 71.0 | 0.40 | 72.5 | 0.82 | 75.8 | 0.52 |
| | TriT-SVM | 92.3 | **0.02** | 91.7 | 0.02 | 83.6 | 0.27 |
| | TriT-DT | 141.5 | **0.0** | 145.2 | **0.0** | 146.1 | **0.0** |
| (DTW) | TriT-1NN | **61.5** | – | 62.7 | 0.89 | 63.1 | 0.66 |
| | SETRED | 65.4 | 0.74 | **61.1** | – | **57.9** | – |
| | Democratic | 71.5 | 0.74 | 70.6 | 0.86 | 71.4 | 0.52 |
| | TriT-SVM | 101.7 | $\mathbf{2.7 \times 10^{-3}}$ | 97.5 | $\mathbf{8.0 \times 10^{-3}}$ | 99.6 | $\mathbf{1.7 \times 10^{-3}}$ |
| | TriT-DT | 139.8 | **0.0** | 148.0 | **0.0** | 147.8 | **0.0** |
| (ACF) | TriT-1NN | **57.9** | – | **63.1** | – | 70.3 | 0.30 |
| | SETRED | 62.3 | 0.71 | 63.3 | 0.98 | **57.8** | – |
| | Democratic | 79.5 | 0.14 | 82.4 | 0.22 | 80.4 | 0.12 |
| | TriT-SVM | 125.4 | **0.0** | 108.3 | $\mathbf{5.6 \times 10^{-4}}$ | 116.8 | $\mathbf{4.0 \times 10^{-6}}$ |
| | TriT-DT | 114.7 | $\mathbf{8.0 \times 10^{-6}}$ | 148.0 | $\mathbf{4.0 \times 10^{-6}}$ | 147.8 | $\mathbf{9.0 \times 10^{-6}}$ |
| (FFT) | Democratic | **64.5** | – | **64.4** | – | **61.9** | – |
| | TriT-1NN | 71.9 | 0.54 | 74.8 | 0.67 | 73.9 | 0.55 |
| | SETRED | 72.9 | 0.54 | 69.5 | 0.67 | 69.0 | 0.55 |
| | TriT-SVM | 90.3 | 0.09 | 86.5 | 0.20 | 87.2 | 0.10 |
| | TriT-DT | 140.1 | **0.0** | 144.6 | **0.0** | 147.8 | **0.0** |
| (ERP) | Democratic | **68.7** | – | **65.4** | – | **71.4** | – |
| | TriT-1NN | 79.4 | 0.71 | 72.0 | 0.83 | 72.3 | 0.98 |
| | SETRED | 73.1 | 0.71 | 68.0 | 0.83 | 71.6 | 0.98 |
| | TriT-SVM | 73.6 | 0.71 | 84.9 | 0.32 | 76.8 | 0.98 |
| | TriT-DT | 145.0 | **0.0** | 149.5 | **0.0** | 147.7 | **0.0** |

Adjusted $p$ values for the post-hoc procedure of Hochberg

both variables present an strong inverse correlation. This means that the presence of overlapping in datasets affects, in a significant manner, the performance of the self-labeled techniques.

Considering the impact of overlapping in the techniques performance, we present a study about the tuning of some parameters based on the level of overlapping in the datasets. Specifically, we study the parameter significance threshold that controls the addition mechanism in the methods SETRED and SNNRCE. In the case of SETRED, this parameter controls the hypothesis used to decide if an specific example must be added or not to the labeled set $L$. The smaller this value, the more restrictive the selection of examples that are considered good is. In the case of SNNRCE, the significance threshold is related to the hypothesis used to determine if an example is considered as a "doubt example." The greater this value, the more examples will be considered as doubt and accordingly to the SNNRCE method they will be relabeled.

**Table 12** Run-time means (s) associated with the training and testing process of the self-labeled techniques for each labeled percent evaluated

| Labeled (%) | SelfT-1NN | SETRED | SNNRCE | TriT-1NN | Democratic | SelfT-SVM | TriT-SVM | SelfT-DT | TriT-DT |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 539.45 | 970.99 | 2144.17 | 8.67 | 68,805.49 | 95,584.17 | 16,749.58 | 8549.26 | 3421.73 |
| 20 | 783.28 | 1342.92 | 2139.18 | 8.87 | 90,215.12 | 109,860.63 | 36,770.91 | 14,051.83 | 3226.10 |
| 30 | 960.12 | 1449.49 | 2141.10 | 9.17 | 115,380.58 | 98,829.78 | 54,327.15 | 7477.80 | 5273.51 |

**Table 13** The proportions of neighbors with different class computed for each training instance, using the information provided by a neighborhood graph

| Datasets | Euc | | | DTW | | | ACF | | | FFT | | | ERP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% |
| CBF | 0.18 | 0.14 | 0.11 | 0.12 | 0.08 | 0.06 | 0.26 | 0.22 | 0.20 | 0.18 | 0.14 | 0.11 | 0.08 | 0.05 | 0.04 |
| Chlorine | 0.49 | 0.43 | 0.38 | **0.50** | 0.45 | 0.40 | **0.51** | 0.46 | 0.41 | 0.49 | 0.43 | 0.38 | **0.50** | 0.45 | 0.39 |
| CinC_ECG_t | 0.18 | 0.12 | 0.08 | 0.17 | 0.11 | 0.09 | 0.20 | 0.15 | 0.12 | 0.18 | 0.12 | 0.08 | 0.23 | 0.14 | 0.12 |
| Coffee | 0.21 | 0.18 | 0.13 | 0.28 | 0.16 | 0.19 | 0.31 | 0.28 | 0.16 | 0.21 | 0.18 | 0.13 | 0.26 | 0.19 | 0.17 |
| Cricket_X | **0.77** | **0.71** | **0.66** | **0.70** | **0.63** | **0.58** | **0.71** | **0.64** | **0.61** | **0.77** | **0.71** | **0.66** | **0.66** | **0.57** | **0.52** |
| Cricket_Y | **0.75** | **0.69** | **0.66** | **0.69** | **0.61** | **0.57** | **0.71** | **0.66** | **0.63** | **0.75** | **0.69** | **0.66** | **0.63** | **0.54** | **0.50** |
| Cricket_Z | **0.77** | **0.69** | **0.66** | **0.71** | **0.62** | **0.59** | **0.69** | **0.63** | **0.60** | **0.77** | **0.69** | **0.66** | **0.67** | **0.56** | **0.52** |
| ECG | 0.06 | 0.04 | 0.03 | 0.06 | 0.05 | 0.04 | 0.08 | 0.05 | 0.04 | 0.06 | 0.04 | 0.03 | 0.05 | 0.04 | 0.03 |
| ECGFiveD | 0.18 | 0.14 | 0.12 | 0.20 | 0.16 | 0.14 | 0.07 | 0.05 | 0.03 | 0.18 | 0.14 | 0.12 | 0.19 | 0.14 | 0.11 |
| FaceAll | 0.46 | 0.36 | 0.33 | 0.31 | 0.24 | 0.22 | 0.43 | 0.35 | 0.32 | 0.46 | 0.36 | 0.33 | 0.28 | 0.22 | 0.20 |
| FacesUCR | 0.47 | 0.37 | 0.33 | 0.32 | 0.25 | 0.22 | 0.43 | 0.35 | 0.31 | 0.47 | 0.37 | 0.33 | 0.28 | 0.22 | 0.20 |
| Fish | **0.53** | 0.46 | 0.40 | **0.52** | 0.44 | 0.40 | **0.55** | 0.49 | 0.46 | **0.53** | 0.46 | 0.40 | 0.49 | 0.41 | 0.35 |
| Gun_Point | 0.35 | 0.27 | 0.21 | 0.25 | 0.19 | 0.14 | 0.28 | 0.23 | 0.19 | 0.35 | 0.27 | 0.21 | 0.26 | 0.19 | 0.15 |
| Haptics | **0.67** | **0.67** | **0.65** | **0.68** | **0.67** | **0.64** | **0.68** | **0.68** | **0.67** | **0.67** | **0.67** | **0.65** | **0.66** | **0.63** | **0.62** |
| InlineS | **0.78** | **0.74** | **0.70** | **0.76** | **0.70** | **0.65** | **0.79** | **0.75** | **0.73** | **0.78** | **0.74** | **0.70** | **0.71** | **0.64** | **0.59** |
| Italy | 0.06 | 0.05 | 0.05 | 0.09 | 0.08 | 0.08 | 0.12 | 0.09 | 0.09 | 0.06 | 0.05 | 0.05 | 0.11 | 0.08 | 0.08 |
| Lighting2 | 0.41 | 0.39 | 0.42 | 0.35 | 0.34 | 0.30 | 0.35 | 0.34 | 0.33 | 0.41 | 0.39 | 0.42 | 0.37 | 0.32 | 0.29 |
| Lighting7 | **0.66** | **0.60** | **0.58** | **0.60** | 0.48 | 0.45 | **0.55** | **0.53** | 0.49 | **0.66** | **0.60** | **0.58** | **0.57** | 0.46 | 0.41 |
| MALLAT | 0.10 | 0.08 | 0.07 | 0.11 | 0.08 | 0.07 | 0.12 | 0.09 | 0.08 | 0.10 | 0.08 | 0.07 | 0.15 | 0.12 | 0.10 |
| MedicalI | **0.51** | 0.45 | 0.41 | **0.52** | 0.44 | 0.40 | **0.56** | 0.48 | 0.44 | **0.51** | 0.45 | 0.41 | 0.49 | 0.41 | 0.39 |
| MoteStrain | 0.14 | 0.14 | 0.14 | 0.13 | 0.11 | 0.10 | 0.25 | 0.23 | 0.22 | 0.14 | 0.14 | 0.14 | 0.13 | 0.11 | 0.11 |
| Olive | 0.34 | 0.33 | 0.25 | 0.34 | 0.32 | 0.27 | 0.40 | 0.37 | 0.30 | 0.34 | 0.33 | 0.25 | 0.38 | 0.33 | 0.27 |
| OSULeaf | **0.67** | **0.62** | **0.59** | **0.67** | **0.61** | **0.58** | **0.51** | 0.46 | 0.45 | **0.67** | **0.62** | **0.59** | **0.65** | **0.59** | **0.55** |

**Table 13** continued

| Datasets | Euc | | | DTW | | | ACF | | | FFT | | | ERP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% | 10% | 20% | 30% |
| Sony | 0.16 | 0.12 | 0.09 | 0.16 | 0.13 | 0.11 | 0.15 | 0.12 | 0.11 | 0.16 | 0.12 | 0.09 | 0.13 | 0.11 | 0.09 |
| SonyII | 0.16 | 0.13 | 0.10 | 0.18 | 0.14 | 0.13 | 0.17 | 0.13 | 0.11 | 0.16 | 0.13 | 0.10 | 0.16 | 0.13 | 0.12 |
| StarLightC | 0.17 | 0.15 | 0.14 | 0.13 | 0.11 | 0.10 | 0.11 | 0.10 | 0.09 | 0.17 | 0.15 | 0.14 | 0.14 | 0.13 | 0.12 |
| Synthetic | 0.41 | 0.35 | 0.31 | 0.32 | 0.24 | 0.22 | 0.41 | 0.39 | 0.37 | 0.41 | 0.35 | 0.32 | 0.20 | 0.16 | 0.14 |
| Trace | 0.49 | **0.50** | 0.48 | 0.39 | 0.33 | 0.26 | 0.36 | 0.31 | 0.26 | 0.49 | **0.50** | 0.48 | 0.35 | 0.34 | 0.32 |
| Two | **0.53** | 0.48 | 0.45 | 0.40 | 0.34 | 0.31 | **0.55** | **0.53** | **0.52** | **0.53** | 0.48 | 0.45 | 0.25 | 0.20 | 0.17 |
| TwoLeadECG | 0.19 | 0.13 | 0.10 | 0.12 | 0.09 | 0.08 | 0.11 | 0.09 | 0.09 | 0.19 | 0.13 | 0.10 | 0.07 | 0.05 | 0.05 |
| uWaveGL_X | 0.40 | 0.36 | 0.34 | 0.39 | 0.36 | 0.33 | **0.56** | **0.55** | **0.53** | 0.40 | 0.36 | 0.34 | 0.37 | 0.34 | 0.32 |
| uWaveGL_Y | 0.44 | 0.41 | 0.39 | 0.44 | 0.41 | 0.39 | **0.66** | **0.64** | **0.63** | 0.44 | 0.41 | 0.39 | 0.44 | 0.42 | 0.40 |
| uWaveGL_Z | 0.45 | 0.41 | 0.40 | 0.44 | 0.40 | 0.38 | **0.55** | **0.53** | **0.52** | 0.45 | 0.41 | 0.40 | 0.42 | 0.38 | 0.37 |
| Wafer | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | 0.01 | 0.02 | 0.02 | 0.01 | 0.02 | 0.01 | 0.01 | 0.03 | 0.02 | 0.01 |
| Yoga | 0.26 | 0.21 | 0.17 | 0.24 | 0.19 | 0.16 | 0.27 | 0.22 | 0.21 | 0.26 | 0.21 | 0.17 | 0.24 | 0.19 | 0.16 |

The values highest than 0.5 were highlighted as bold

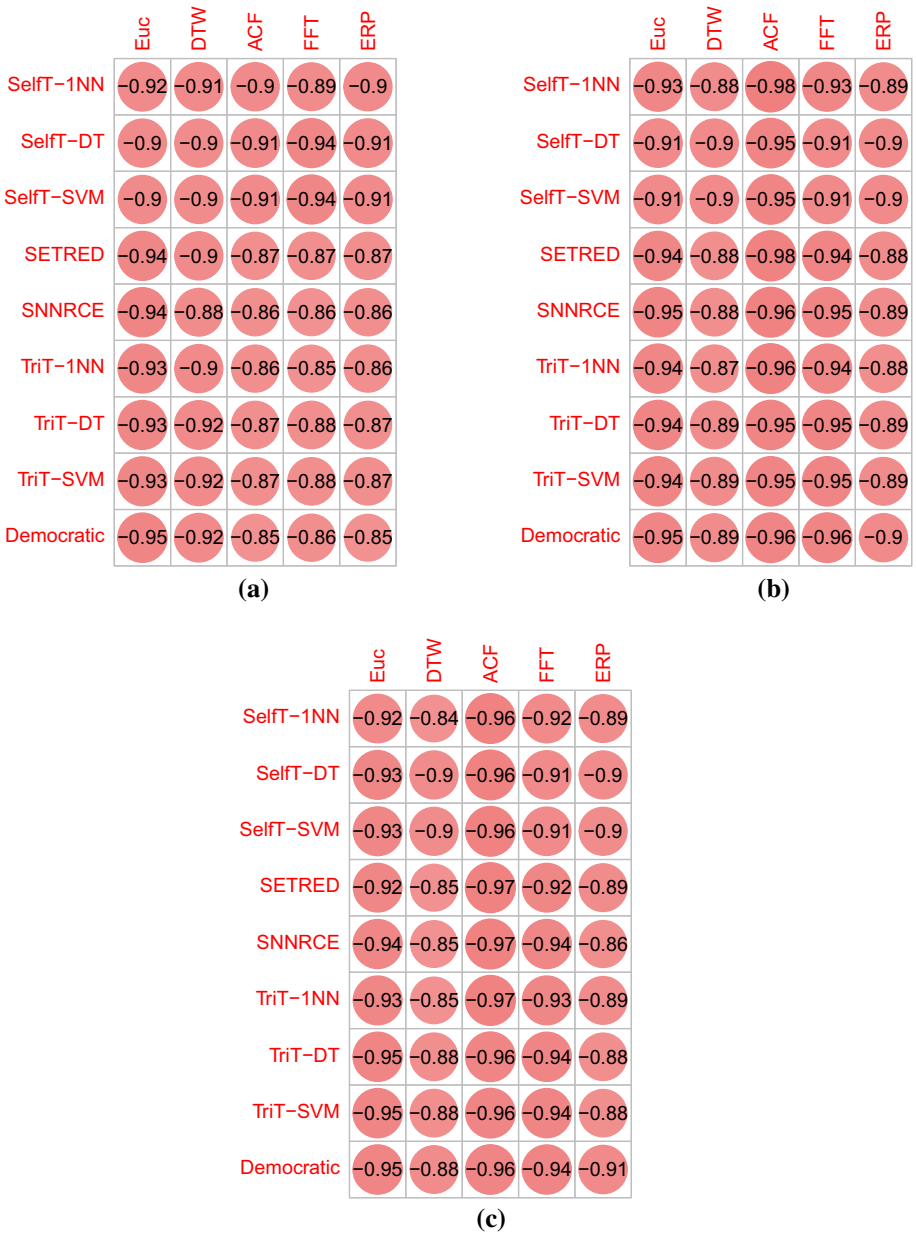



**Fig. 5** Spearman's $\rho$ correlation coefficients obtained between the overlapping estimated in the datasets and the accuracy results of the self-labeled techniques. **a** 10% labeled data. **b** 20% labeled data. **c** 30% labeled data

Figures 6 and 7 show the behavior of the significance threshold parameter throughout different levels of overlap. The significance threshold selected is the value, between three possible values (0.05, 0.10 and 0.15), that maximizes the accuracy of SETRED. In general, the most appropriated threshold for datasets seems to be the most restrictive value 0.05. This

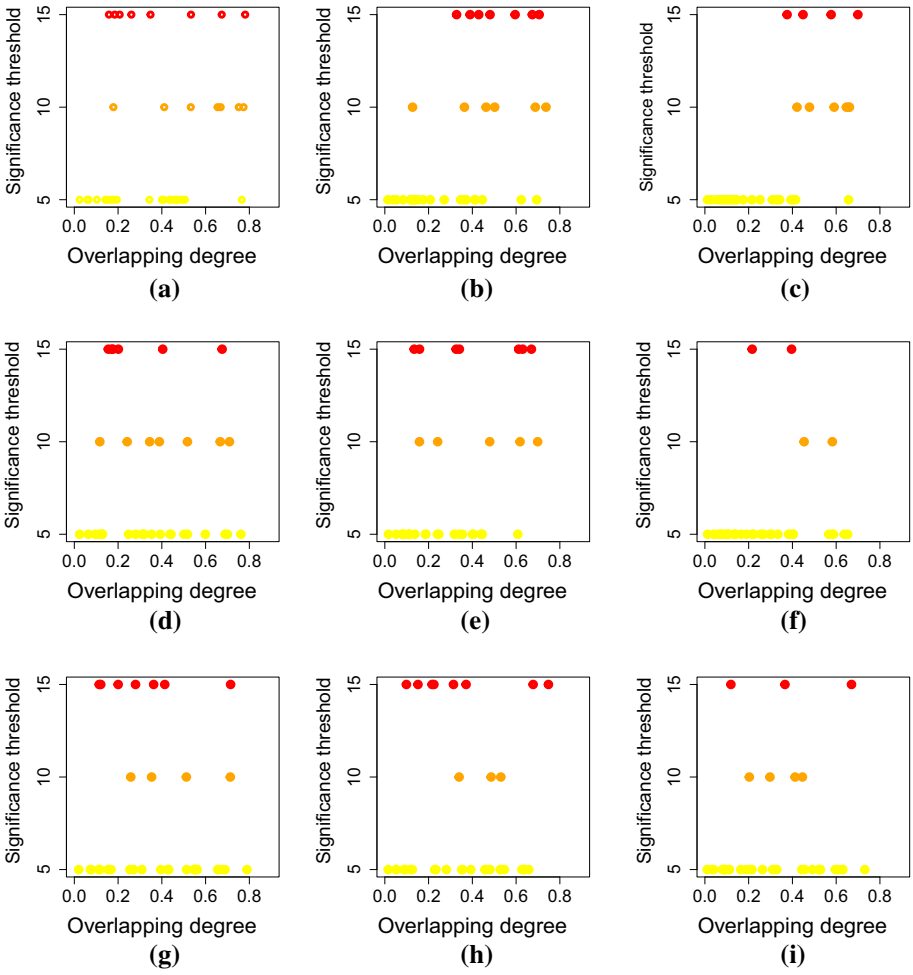



**Fig. 6** Best configuration of the significance threshold parameter for each dataset in the SETRED method. **a** Euclidean 10%, **b** Euclidean 20%, **c** Euclidean 30%, **d** DTW 10%, **e** DTW 20%, **f** DTW 30%, **g** ACF 10%, **h** ACF 20% and **i** ACF 30%

value benefits datasets with low overlapping. For other datasets, including those with medium or high degree of overlap are more flexible values of significance threshold preferred. This behavior is more noticeable at 30% of labeled data.

Figures 8 and 9 show a similar behavior for the SNNRCE method. For this method coincides as the most suitable option for the significance threshold the value 0.05. Although, for some datasets with more overlapping, the values 0.10 and 0.15 result a better option. In contrast to SETRED, this behavior is more noticeable at 10% of labeled data.

### 4.5 Can semi-supervised learning improve classification performance?

A recommendable procedure [12] in presence of labeled and unlabeled data is to start by learning a supervised classifier from the labeled data, named "baseline classifier." A compar-

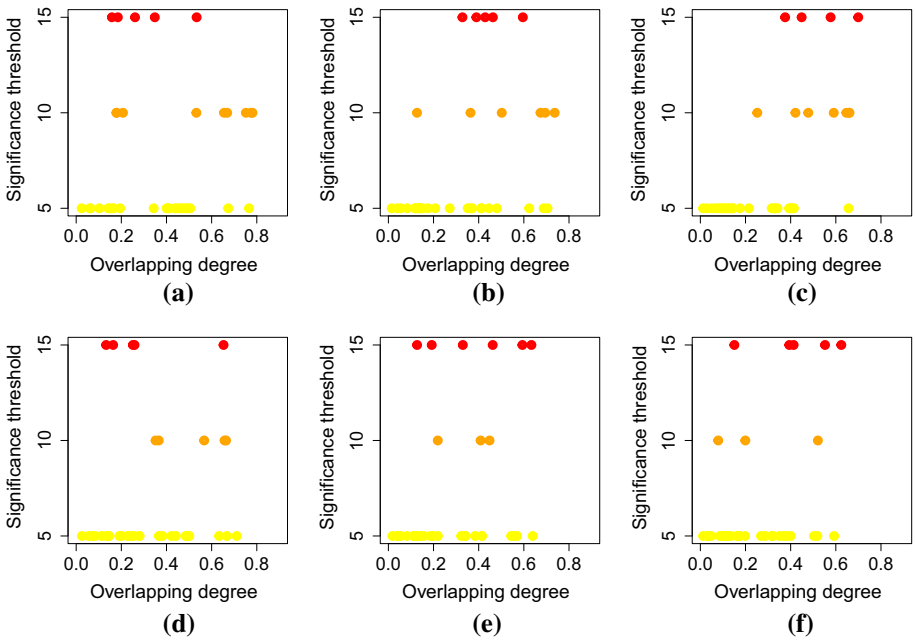



**Fig. 7** Best configuration of the significance threshold parameter for each dataset in the SETRED method. **a** FFT 10%, **b** FFT 20%, **c** FFT 30%, **d** ERP 10%, **e** ERP 20% and **f** ERP 30%

ison with this classifier allows us to identify situations where the addition of unlabeled data causes performance degradation of the classifier obtained. In this section, we perform such analysis measuring the accuracy gain obtained with the addition of unlabeled data during the training phase. We estimate the accuracy gain subtracting the accuracy obtained with supervised classification from the accuracy obtained with SSC. In both cases, the classifier performance is evaluated on the testing set, using the same fivefold cross-validation scheme. We select as the baseline method the 1NN classifier because it offers the most accurate results.

We expect that the best performing self-labeled methods selected from previous sections will obtain the highest accuracy gain. For this reason, we focus this analysis on those methods. Figure 10 shows the accuracy gain obtained for each dataset using three semi-supervised methods. A negative gain means performance degradation of the classifier. We can observe a very diverse behavior of the gains under the different labeled ratios and methods. There are datasets that do not benefit from SSL, for instance ECG [60] and Wafer. The size of these datasets already causes that the hypothesis learned by the supervised baseline is perfectly capable of obtaining accurate classification results in the inductive phase. On other datasets, such as MedicalI, classification performance deteriorates with the addition of unlabeled data for 10 and 20% labeled ratio. This is the case as the initial labeled data provided are insufficient for training a correct model where unlabeled data will be truly beneficial. It is noticeable that MedicalI is a multi-class dataset (10 classes) with high overlapping. This adverse situation starts to reverse at 30% labeled ratio where Democratic obtains a positive accuracy difference gain.

Though there are unfavorable situations for SSL in some datasets, Fig. 11 shows the gains obtained with SETRED, in decreasing order. In general, at 20% labeled data a significant

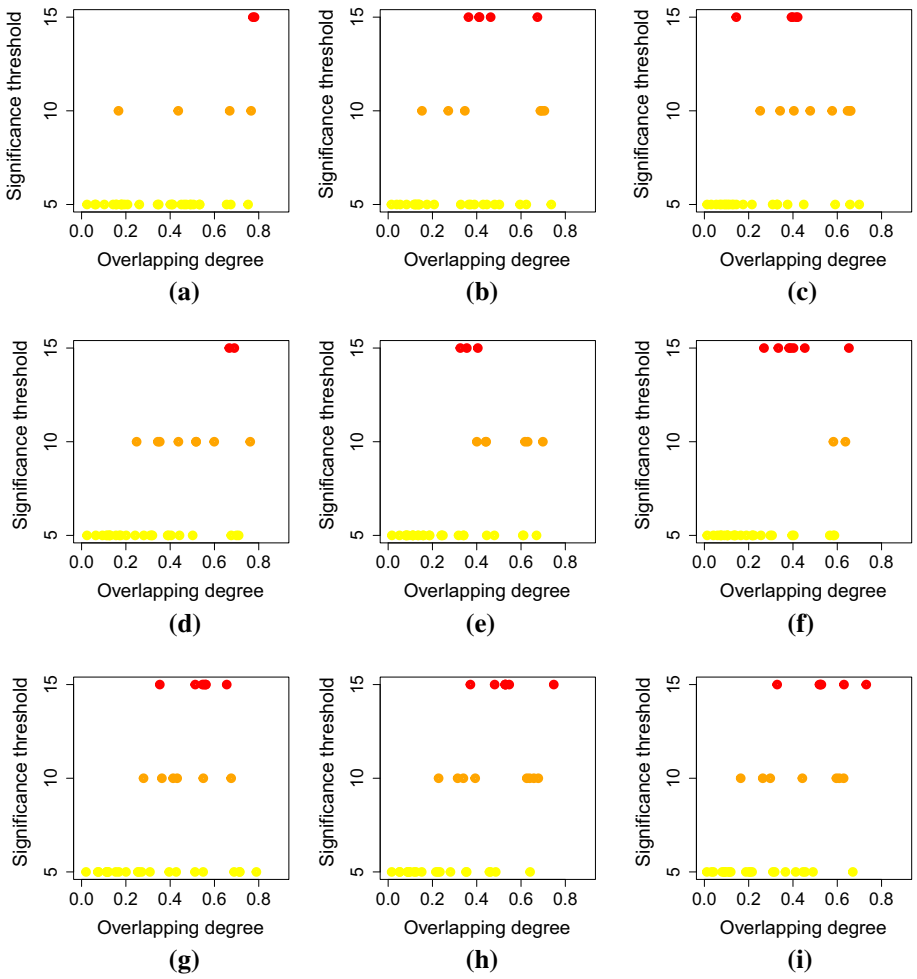



**Fig. 8** Best configuration of the significance threshold parameter for each dataset in the SNNRCE method. **a** Euclidean 10%, **b** Euclidean 20%, **c** Euclidean 30%, **d** DTW 10%, **e** DTW 20%, **f** DTW 30%, **g** ACF 10%, **h** ACF 20% and **i** ACF 30%

positive gain can be observed. We also see that it depends heavily on the ratio of labeled instances if there is benefit and how big it is. Summarizing, the circumstances that make SSL a suitable approach for a particular dataset depend on the modeling assumptions adopted for the classifier, as well as the characteristics of the time series data.

In addition to the performed analysis, we consider the baseline classifier trained on the fully labeled training set and evaluated on the testing set. These accuracy results can be considered as an upper bound for the self-labeled methods. Figure 12 shows the semi-supervised classification accuracy bounded by the baseline classifier. In general, the semi-supervised results in most of the datasets are competitive compared with the upper bound considered. Interestingly, for datasets such as StarLightC and Synthetic the multi-learning hypothesis, learned by Democratic, outperforms the upper bound classifier.

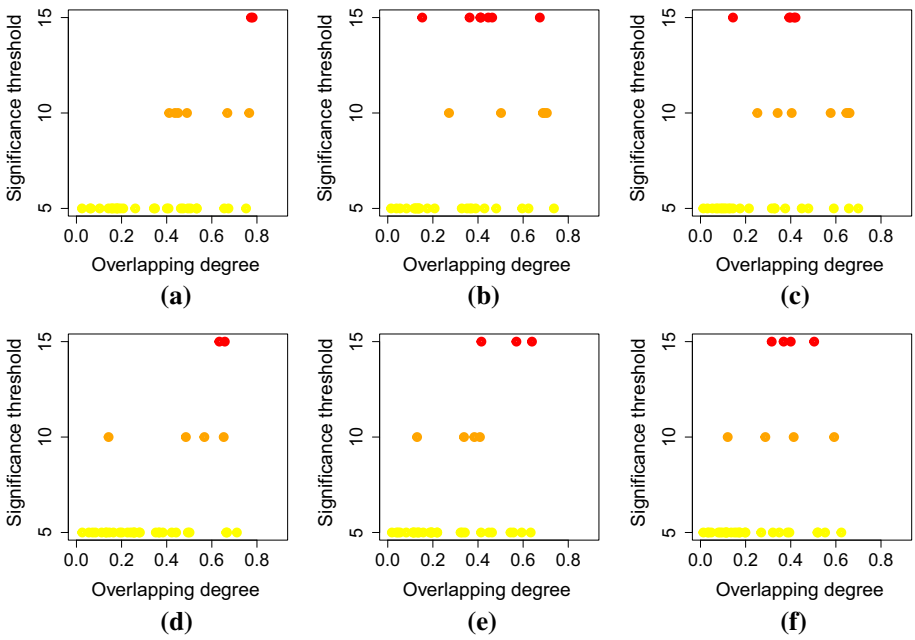



**Fig. 9** Best configuration of the significance threshold parameter for each dataset in the SNNRCE method. **a** FFT 10%, **b** FFT 20%, **c** FFT 30%, **d** ERP 10%, **e** ERP 20% and **f** ERP 30%

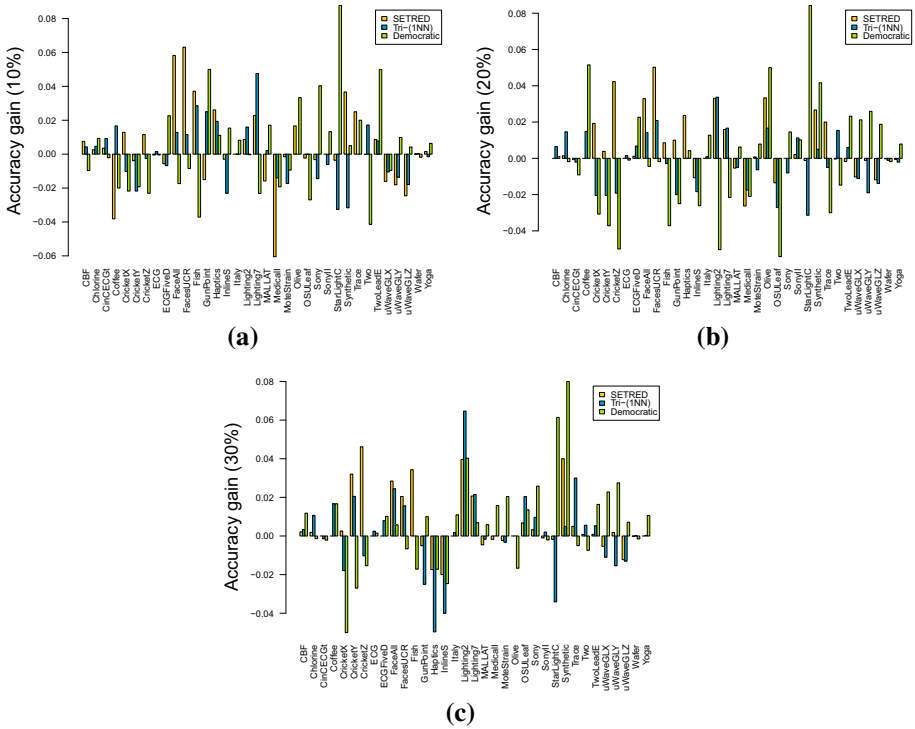



**Fig. 10** *Bar plot* of the accuracy gain for the best performing methods in inductive phase, using Euclidean distance. **a** 10% labeled data. **b** 20% labeled data. **c** 30% labeled data

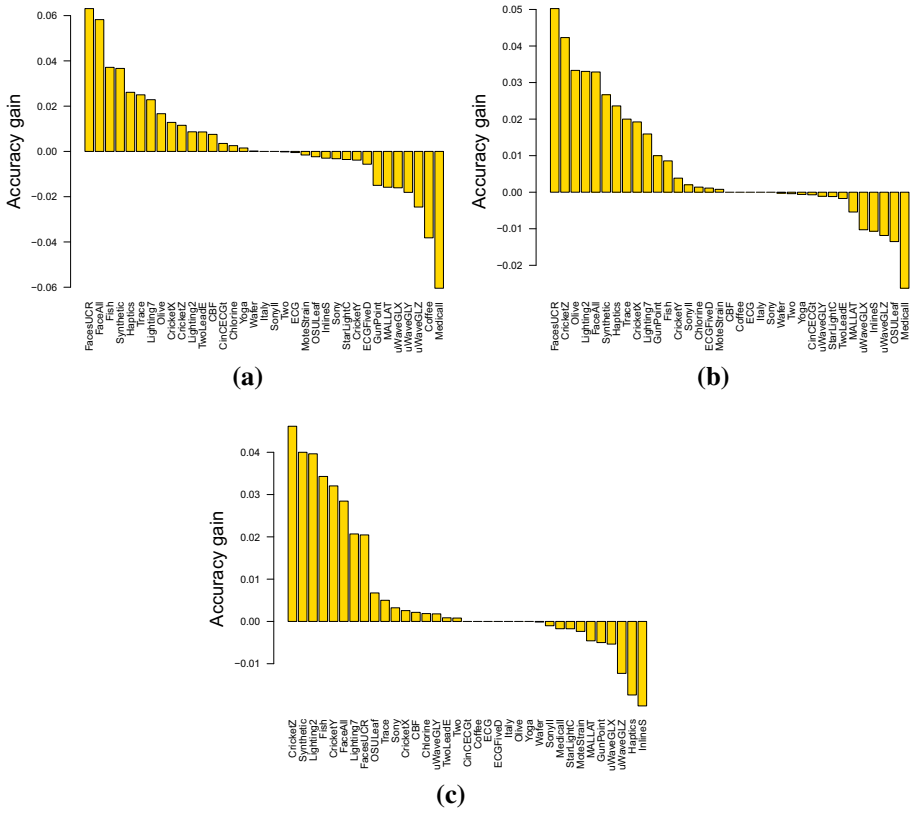



**Fig. 11** *Bar plot* of the accuracy gain for SETRED in inductive phase, using Euclidean distance. **a** 10% labeled data. **b** 20% labeled data. **c** 30% labeled data

## 4.6 General discussion

This section gives a general discussion of the properties observed throughout this study. In addition, we highlight the methods that perform best in general.

- For most of the methods, accuracy increases with an increase of labeled examples. However, this increase is usually rather moderate in most methods. In self-labeled methods with SVMs as base classifiers, the increase is bigger than in the other methods.
- The classical SelfT is clearly outperformed by other self-labeled techniques independently of the learning approach and the dissimilarity measure used.
- Usually, there is no difference between the obtained rankings with accuracy and kappa statistics. This means that there is no significant difference in the way that the classifiers benefit from random hits.
- In general, 1NN offers the best transductive and inductive results as base classifier in most self-labeled methods. In addition, this base classifier yields the most competitive run-times in comparison with SVM and DT.
- Although SVM and DT do not offer competitive results as base classifiers, when these learning schemes are combined with 1NN, following a multi-learning scheme (Demo-

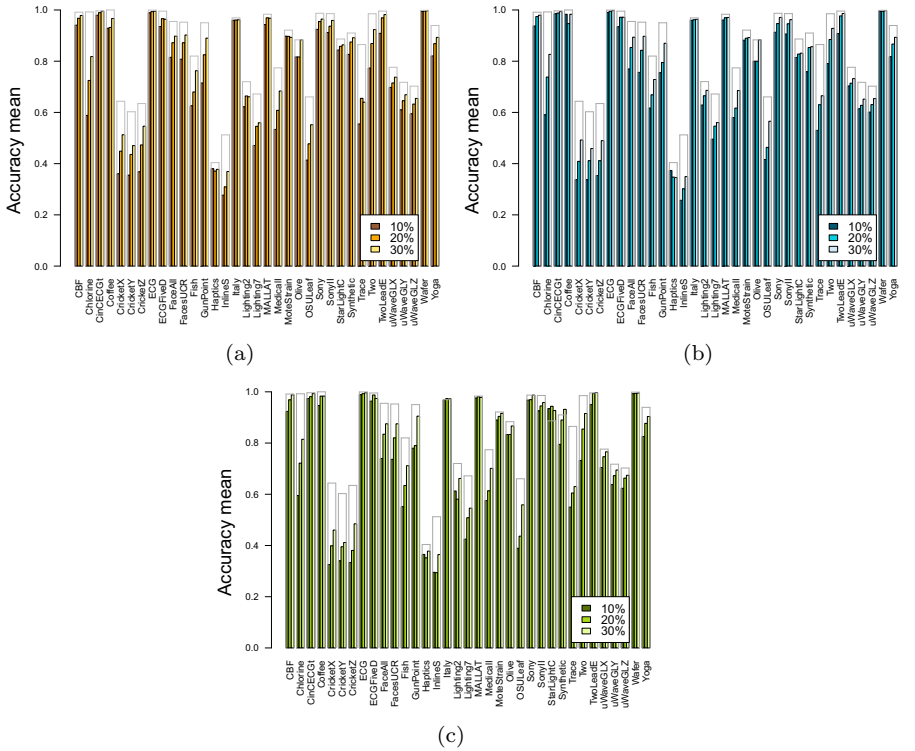



**Fig. 12** How far removed is semi-supervised from supervised results, using inductive learning under Euclidean distance. The *upper gray lines* represent the values of the baseline classifier trained on the fully labeled training set. **a** SETRED. **b** TriT-1NN. **c** Democratic

cratic), good results are obtained. However, the increment of the run-time is a side effect of the Democratic method.

- The use of DTW and ERP distance results in a gain of accuracy in comparison with the other measures studied. In the case of DTW, this advantage is reduced in presence of SVM as a base classifier. This behavior is caused by the indefiniteness of the kernels constructed under DTW.
- SETRED, Democratic and TriT-1NN are the best performing methods in this study. TriT-SVM also exhibits a competitive behavior under FFT and ERP dissimilarity measures. For Euclidean, ERP and FFT, we recommend the use of Democratic. For DTW, we recommend TriT-1NN and SETRED for transductive and inductive scenarios, respectively. For ACF, we recommend TriT-1NN for either inductive and transductive learning.
- The overlapping in datasets is an aspect that should be taken into account during the solution of time series classification problems. We find strong evidence about the negative effects of overlap in the accuracy of the self-labeled techniques.
- From the study of the significance threshold parameter, we recommend in general the use of the most restrictive value 0.05 in both methods SETRED and SNNRCE. In particular, for datasets with high levels of overlap, other values of this parameter must be considered.

## 5 Conclusion

We have investigated the applicability of different self-labeled methods for semi-supervised learning in a time series context. In addition to the popular SelfT with 1NN as base learner, we have explored other combinations of self-labeled methods and learning schemes that had not been applied in a time series context, to the best of our knowledge. We can conclude with the following remarks:

- In general, 1NN is a robust choice for the base classifier in the semi-supervised context as it offers the most accurate results and no parameters have to be tuned.
- SelfT is always outperformed by other self-labeled methods such as TriT and SETRED.
- Our empirical study allows us to highlight three methods, in particular SETRED, TriT-1NN and Democratic, that perform significantly better than the rest in terms of transductive and inductive capabilities.
- The use of ensembles of classifiers (TriT-1NN and Democratic) is a very promising attempt to perform SSL in the time series context. This is in line with recent studies [3,39] in supervised classification of temporal data.
- Taking into account the underlying risk to classification performance caused by the addition of unlabeled data, we recommend a comparison of the SSL results with the 1NN as baseline classifier to identify the real benefits of learning with unlabeled data. The overlapping in datasets is other aspect that should be taken into account in the selection of the classification techniques. Specifically, the performance of the self-labeled techniques can be seriously affected by the presence of overlapping.

**Acknowledgements** We thank anonymous reviewers for their very useful comments and suggestions. This work was supported in part by "Proyecto de Investigación de Excelencia de la Junta de Andalucía, P12-TIC-2958," "Proyecto de Investigación del Ministerio de Economía y Competitividad, TIN2013-47210-P" and TIN-2016-81113-R. This work was partly performed while M. González held a travel grant from the Asociación Iberoamericana de Postgrado (AUIP), supported by Junta de Andalucía, to undertake a research stay at University of Granada.

## References

1. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6(1):37–66
2. Bagnall AJ, Janacek GJ (2004) Clustering time series from ARMA models with clipped data. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining', KDD '04. ACM, New York, NY, pp49–58
3. Bagnall A, Lines J, Hills J, Bostrom A (2015) Time-series classification with COTE: the collective of transformation-based ensembles. IEEE Trans Knowl Data Eng 27(9):2522–2535
4. Balakrishnan S, Madigan D (2006) Decision trees for functional variables. In: Sixth international conference on data mining, ICDM '06, pp 798–802
5. Batista G, Hao Y, Keogh E, MafraNeto A (2011) Towards automatic classification on flying insects using inexpensive sensors. In: IEEE 10th international conference on machine learning and applications (ICMLA). IEEE, vol 1, pp 364–369
6. Begum N, Hu B, Rakthanmanon T, Keogh E (2014) A minimum description length technique for semi-supervised time series classification. In: Integration of reusable systems, vol 263 of advances in intelligent systems and computing. Springer, Berlin, pp 171–192
7. Behera H, Dash P, Biswal B (2010) Power quality time series data mining using S-transform and fuzzy expert system. Appl Soft Comput 10(3):945–955
8. Ben-David A (2007) A lot of randomness is hiding in accuracy. Eng Appl Artif Intell 20(7):875–885
9. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Eleventh annual conference on computational learning theory, COLT' 98. ACM, New York, NY, pp 92–100
10. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

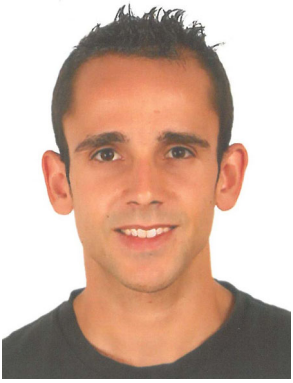
11. Carden EP, Brownjohn JM (2008) ARMA modelled time-series classification for structural health monitoring of civil infrastructure. Mech Syst Signal Process 22(2):295–314
12. Chapelle O, Schölkopf B, Zien A (2006) Semi-supervised learning, vol 2. MIT Press, Cambridge
13. Chen L, Özsu MT, Oria V (2005) Robust and fast similarity search for moving object trajectories, In: Proceedings of the 2005 ACM SIGMOD international conference on management of data, SIGMOD '05. ACM, New York, NY, pp 491–502
14. Chen Y, Hu B, Keogh E, Batista GE (2013) DTW-D: time series semi-supervised learning from a single example. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '13. ACM, New York, NY, pp 383–391
15. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The ucr time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/
16. Cuturi M (2011) Fast global alignment kernels. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 929–936
17. Dash P, Behera H, Lee I (2008) Time sequence data mining using time-frequency analysis and soft computing techniques. Appl Soft Comput 8(1):202–215
18. De Sousa CAR, Souza VMA, Batista GEAPA (2014) Time series transductive classification on imbalanced data sets: an experimental study. In: 22nd international conference on pattern recognition (ICPR), pp 3780–3785
19. De Sousa CAR, Souza VMA, Batista GEAPA (2015) An experimental analysis on time series transductive classification on graphs. In: International joint conference on neural networks (IJCNN), pp 1–8
20. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
21. Douzal-Chouakria A, Amblard C (2012) Classification trees for time series. Pattern Recogn 45(3):1076–1091
22. Faloutsos, C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. In: Proceedings of the 1994 ACM SIGMOD international conference on management of data, SIGMOD '94. ACM, New York, NY, pp 419–429
23. Flesca S, Manco G, Masciari E, Pontieri L, Pugliese A (2007) Exploiting structural similarity for effective web information extraction. Data Knowl Eng 60(1):222–234
24. Frank J, Mannor S, Pineau J, Precup D (2013) Time series analysis using geometric template matching. IEEE Trans Pattern Anal Mach Intell 35(3):740–754
25. Fu T (2011) A review on time series data mining. Eng Appl Artif Intell 24(1):164–181
26. Fulcher BD, Jones NS (2014) Highly comparative feature-based time-series classification. IEEE Trans Knowl Data Eng 26(12):3026–3037
27. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180(10):2044–2064
28. Geler Z, Kurbalija V, Radovanović M, Ivanović M (2015) Comparison of different weighting schemes for the kNN classifier on time-series data. Knowl Inf Syst 48:331–378
29. Goldman SA, Zhou Y (2000) Enhancing supervised learning with unlabeled data. In: Seventeenth international conference on machine learning (ICML), pp 327–334
30. González M, Bergmeir C, Triguero I, Rodríguez Y, Benítez JM (2016) On the stopping criteria for k-nearest neighbor in positive unlabeled time series classification problems. Inf Sci 328:42–59
31. Hochberg Y, Rom D (1995) Extensions of multiple testing procedures based on Simes' test. J Stat Plan Inference 48(2):141–152
32. Hodges J, Lehmann EL et al (1962) Rank methods for combination of independent experiments in analysis of variance. Ann Math Stat 33(2):482–497
33. Jeong Y, Jayaraman R (2015) Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification. Knowl-Based Syst 75:184–191
34. Kaya H, GunduzOguducu S (2015) A distance based time series classification framework. Inf Syst 51:27–42
35. Kim M (2013) Semi-supervised learning of hidden conditional random fields for time-series classification. Neurocomputing 119:339–349
36. Kurbalija V, Radovanović M, Geler Z, Ivanović M (2014) The influence of global constraints on similarity measures for time-series databases. Knowl-Based Syst 56:49–67
37. Lei H, Sun B (2007) A study on the dynamic time warping in kernel machines. In: Third international IEEE conference on signal-image technologies and internet-based system (SITIS), SITIS '07, pp 839–845
38. Li M, Zhou Z (2005) Setred: self-training with editing. In: Advances in knowledge discovery and data mining, vol 3518 of Lecture notes in computer science. Springer, Berlin, pp 611–621
39. Lines J, Bagnall A (2014) Time series classification with ensembles of elastic distance measures. Data Min Knowl Discov 29(3):565–592

40. Liu Y, Yao K, Liu S, Raghavendra CS, Balogun O and Olabinjo L (2011) Semi-supervised failure prediction for oil production wells. In: IEEE 11th international conference on data mining workshops (ICDMW). IEEE, pp 434–441
41. Marteau PF (2009) Time warp edit distance with stiffness adjustment for time series matching. IEEE Trans Pattern Anal Mach Intell 31(2):306–318
42. Marteau P, Gibet S (2015) On recursive edit distance kernels with application to time series classification. IEEE Trans Neural Netw Learn Syst 26(6):1121–1133
43. Marussy K, Buza K (2013) Success: a new approach for semi-supervised classification of time-series. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh L, Zurada J (eds) Artificial intelligence and soft computing, vol 7894. Lecture notes in computer science. Springer, Berlin, pp 437–447
44. Meng J, Wu L, Wang X, Lin T (2011) Granulation-based symbolic representation of time series and semi-supervised classification. Comput Math Appl 62(9):3581–3590
45. Petitjean F, Forestier G, Webb GI, Nicholson AE, Chen Y, Keogh E (2016) Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowl Inf Syst 47(1):1–26
46. Povinelli R, Johnson M, Lindgren A, Ye J (2004) Time series classification using gaussian mixture models of reconstructed phase spaces. IEEE Trans Knowl Data Eng 16(6):779–783
47. Pree H, Herwig B, Gruber T, Sick B, David K, Lukowicz P (2014) On general purpose time series similarity measures and their use as kernel functions in support vector machines. Inf Sci 281:478–495
48. R Core Team (2016) *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
49. Rabiner L (1989) A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
50. Rakthanmanon T, Campana B, Mueen A, Batista G, Westover B, Zhu Q, Zakaria J, Keogh E (2012) Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '12. ACM, New York, NY, pp 262–270
51. Ratanamahatana CA, Wanichsan D (2008) Stopping criterion selection for efficient semi-supervised time series classification. In: Lee R (ed) Software engineering, artificial intelligence, networking and parallel/distributed computing, vol 149. Studies in computational intelligence. Springer, Berlin, pp 1–14
52. Rodríguez JJ, Alonso CJ (2004) Interval and dynamic time warping-based decision trees. In: Proceedings of the 2004 ACM symposium on applied computing, SAC '04. ACM, pp 548–552
53. Rodríguez JJ, Alonso CJ, Boström H (2000) Learning first order logic time series classifiers: rules and boosting. In: Principles of data mining and knowledge discovery, vol 1910 of Lecture notes in computer science. Springer, Berlin, pp 299–308
54. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49
55. Serrà J, Arcos JL (2014) An empirical evaluation of similarity measures for time series classification. Knowl-Based Syst 67:305–314
56. Shimodaira H, Noma K, Nakai M, Sagayama S (2001) Dynamic time-alignment kernel in support vector machine. In: Proceedings of the 14th international conference on neural information processing systems: natural and synthetic, NIPS'01. MIT Press, Cambridge, MA, pp 921–928
57. Triguero I, García S, Herrera F (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl Inf Syst 42(2):245–284
58. Wang X, Mueen A, Ding H, Trajcevski G, Scheuermann P, Keogh E (2013) Experimental comparison of representation methods and distance measures for time series data. Data Min Knowl Discov 26(2):275–309
59. Wang Y, Xu X, Zhao H, Hua Z (2010) Semi-supervised learning based on nearest neighbor rule and cut edges. Knowl-Based Syst 23(6):547–554
60. Wei L (2006) Datasets used for experimental evaluation in the paper: semi-supervised time series classification. www.cs.ucr.edu/~wli/selfTraining/
61. Wei L, Keogh E (2006) Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 748–753
62. Weng X, Shen J (2008) Classification of multivariate time series using two-dimensional singular value decomposition. Knowl-Based Syst 21(7):535–539
63. Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, third, edition edn. Morgan Kaufmann, Boston
64. Xi X, Keogh E, Shelton C, Wei L, Ratanamahatana CA (2006) Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on machine learning, ICML '06. ACM, New York, pp 1033–1040

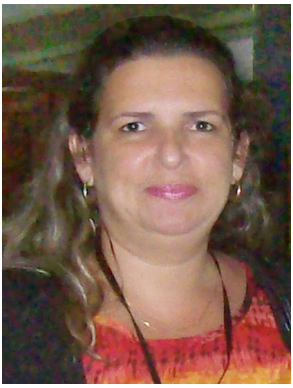
65. Xing Z, Pei J, Yu PS (2012) Early classification on time series. Knowl Inf Syst 31(1):105–127
66. Yamada Y, Suzuki E, Yokoi H, Takabayashi K (2003) Decision-tree induction from time-series data based on a standard-example split test. In: Twentieth international conference on machine learning, vol 3 of ICML '03, pp 840–847
67. Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd annual meeting on association for computational linguistics. Association for Computational Linguistics, pp 189–196
68. Zhang D, Zuo W, Zhang D, Zhang H (2010) Time series classification using support vector machine with Gaussian elastic metric kernel. In: 20th international conference on pattern recognition (ICPR), ICPR '10, pp 29–32
69. Zhou Y, Goldman S (2004) Democratic co-learning. In: IEEE 16th international conference on tools with artificial intelligence (ICTAI). IEEE, pp 594–602
70. Zhou Z, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 17(11):1529–1541
71. Zighed DA, Lallich S, Muhlenbach F (2002) Principles of data mining and knowledge discovery: 6th European conference. In: Separability index in supervised learning, PKDD 2002 Helsinki, Finland, August 19–23, 2002 Proceedings. Springer, Berlin, pp 475–487

**Mabel González** received the M.Sc. degree in Computer Science from the Universidad Central "Marta Abreu" de las Villas, Cuba, in 2010, and the Ph.D. degree from the University of Granada, Spain, in 2016. She is currently working at Department of Computer Science, Universidad Central "Marta Abreu" de las Villas, Villa Clara, Cuba. Her research interests include data mining, semi-supervised learning and time series classification.

**Christoph Bergmeir** received the M.Sc. degree in Computer Science from the University of Ulm, Germany, in 2008, and the Ph.D. degree from the University of Granada, Spain, in 2013. He is currently working at Faculty of Information Technology, Monash University, Melbourne, Australia. He has published in journals such as IEEE Transactions on Neural Networks and Learning Systems, Journal of Statistical Software, Computer Methods and Programs in Biomedicine, and Information Sciences.



**Isaac Triguero** received the M.Sc. and Ph.D. degree in Computer Science from the University of Granada, Spain, in 2009 and 2014, respectively. He is currently an assistant professor in data science at the University of Nottingham, UK. He has published more than 25 papers in international journals. His research interests include data mining, data reduction, evolutionary algorithms, semi-supervised learning, bioinformatics and big data learning.



**Yanet Rodríguez** received the M.Sc. and Ph.D. degree in Computer Science from the Universidad Central "Marta Abreu" de las Villas, Cuba, in 2000 and 2007, respectively. She has published in journals such as International Journal of Hybrid Intelligent Systems as well as more than 12 contributions to conferences. Her research interests include data mining, fuzzy systems, neural networks, data stream and time series classification.

**José Manuel Benítez** received the M.S. and Ph.D. degrees in Computer Science both from the Universidad de Granada, Spain. He is currently an Associate Professor at the Department of Computer Science and Artificial Intelligence, Universidad de Granada. He is the head of the Distributed Computational Intelligence and Time Series (DiCITS) laboratory. His research interests include Cloud Computing and Big Data, Data Science, Computational Intelligence and Time Series. He has published in the leading journals of the "Artificial Intelligence" and Computer Science field. He has led a number of research projects funded by different international and national organizations as well as research contracts with leading international corporations.