

Semi-supervised Time Series Classification Through Image Representations

Bionda Rozin(✉), Emílio Bergamim, Daniel Carlos Guimarães Pedronette, and Fabricio Aparecido Breve

Department of Statistics, Applied Mathematics and Computing (DEMAC). Sao Paulo State University (UNESP), Rio Claro, Brazil
bionda.rozin@unesp.br

Abstract. Time series data is of crucial importance in different domains, such as financial and medical applications. However, obtaining a large amount of labeled time series data is an expensive and time-consuming task, which becomes the process of building an effective machine learning model a challenge. In these scenarios, algorithms that can deal with reduced amounts of labeled data emerge. One example is Semi-Supervised Learning (SSL), which has the capability of exploring both labeled and unlabeled data for tasks such as classification. In this work, a kNN graph-based transductive SSL approach is used for time series classification. A feature extraction step, based on imaging time series and obtaining features using deep neural networks is performed before the classification step. An extensive evaluation is conducted over four datasets, and a parametric analysis of the nearest neighbors is performed. Also, a statistical analysis over the obtained distances is conducted. Results suggest that our methods are suitable for classification and competitive with supervised baselines in some datasets.

Keywords: Transductive Semi Supervised Learning · Graph · Time Series · Feature Extraction · Images · Neural Networks · Classification

1 Introduction

One of the greatest challenges, nowadays, is dealing with great amounts of available data. While there are huge quantities of data available, labeling it, on the other hand, is a laborious, expensive, and time-consuming task. In particular, for time series, finding annotations for data can be difficult due to the necessity of a professional or the excessive amount of instances in a dataset [47].

Time series are important in lots of different fields, such as finances [33], healthcare and medicine [42], energy [2], agriculture [34], and sensor data [12], among others. Given the importance of time series and the scarcity of labeled data, it is important to find methods to process these data without losing effectiveness.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13957, pp. 48–65, 2023.
https://doi.org/10.1007/978-3-031-36808-0_4

For traditional machine learning methods, a large amount of labeled data is necessary for training [47], so, in this scope, Semi-Supervised Learning (SSL) algorithms emerged as an interesting research area, using both labeled and unlabeled information to perform tasks such as classification [41]. Tasks such as disease monitoring [17], human activity recognition [35], indexing of handwritten documents [47], among others, benefit from applying semi-supervised approaches.

In this work, we use a classic transductive SSL algorithm in the task of time series classification. Known as Label Propagation, this method represents the whole dataset as a graph and then labels are propagated from labeled samples to unlabeled ones. Under this scenario, the construction of the similarity graph among pairs of samples is crucial for correct inference.

Under these circumstances, we propose transforming a time series to an image using different methods and subsequently extracting features from the images using neural networks. This step aims to improve the quality of the similarity graph and, consequently, the effectiveness of the classification task.

Our main contributions are:

- Studying different approaches for imaging of time series, extracting features from generated images using neural networks originally trained on a large image dataset, and then using the obtained features to construct a similarity graph for SSL.
- Our classification results show that the proposed methods are suitable for time series classification. We also observe that the imaging method impact results in a more meaningful way than the feature extraction methods.
- A statistical analysis of distances and similarities generated via the proposed method in order to understand classification results.

The rest of the paper is organized as follows: Sect. 2, is given a theoretical foundation about SSL and the related work; Sect. 3 gives a detailing of the proposed method; Sect. 4 outlines the experimental protocol and the used datasets; Sect. 5 presents the obtained results and a discussion; finally, Sect. 6 presents a conclusion about the work.

2 Theoretical Foundation and Related Work

Let $D = \{\mathbf{x}\}_{i=1}^{N}$ be a set of features such that $\mathbf{x_i} \in \mathbb{R}^n$ and $\mathcal{Y} = \{y_i\}_{i=1}^{L}$ a set of labels for the corresponding elements of D, with $L \leq N$. A semi-supervised classification problem appears when one has $L << N$ and wishes to determine labels for the whole dataset (in a transductive setting) or to learn a classification function $y(\mathbf{x})$ (which is called an inductive setting) [41].

Transductive Semi-Supervised Learning (SSL) algorithms are essentially graph-based and can be described as having two steps [41]:

- **Graph construction**: in this step, elements of D are mapped to nodes in a graph $G = (V, E)$, where V is the node set and E the edge set. Weighted edges between similar nodes are drawn in order to describe class relations.

– **Inference**: a dynamic is then applied to G in order to label the unlabelled elements through the previous knowledge present in $\mathcal{Y}$.

The construction of G is crucial to the success of the inference step. Edges in G should connect mainly nodes belonging to the same class, in order to enhance the contrast between different classes. This is related to the clustering hypothesis, which establishes that classes should be associated to clusters in order for SSL to take place [6]. Therefore, we proceed to a deeper discussion of this step.

2.1 Similarity Construction

Letting $d(\cdot, \cdot)$ be a distance function, a common approach to weight edges of G is the Gaussian similarity:

$$w_{i,j} = \exp\left\{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right\} \tag{1}$$

and d is usually set to be the Euclidean distance [39,41]

After weighting, a sparsification procedure is applied [41]. Starting from the fully connected weighted graph edges between less similar nodes are deleted (which is equivalent to setting $w_{i,j} = 0$). In this work, we will use a $k-$nearest neighbors approach, where each node is connected to its k nearest neighbors according to the distance d. This is a common approach in the field and setting $k = \log_2 N$ is known to be an effective choice [3,39].

Tunning of σ is also a difficult problem [41]. However, a comparison of different approaches found that the method proposed in [22] yields better results. In this case, the parameter is set to

$$\sigma = \frac{1}{3N}\sum_{i=1}^{N} d(\mathbf{x}_i, \mathbf{x}_{i_k}), \tag{2}$$

where i_k is the $k-$nearest neighbor of the $i-$th element.

2.2 Label Propagation

The dynamics for classification on G usually involve the propagation of labels from labeled nodes to unlabeled ones. In this direction, *Gaussian Random Fields* (GRF) [41,49] is a well-established method in the SSL community, together with *Local And Global Consistency* [48]. It is also noteworthy that methods that do not rely on the said dynamics exist, such as neural network based [4,24] and biologically inspired methods [5].

We will focus on GRF, due to its widespread use. This approach amounts to the minimization of the cost function

$$H = -\sum_{i<j} w_{i,j} \sum_{s} \psi_{i,s}\psi_{j,s}, \tag{3}$$

where $\psi_{i,s}$ denotes the probability of the $i-$th node belonging to the $s-$th class and $w_{i,j}$ is the pairwise similarity among nodes.

Minimization of H is conducted under constraints on the probabilities of labeled nodes. If $i \leq L$, probabilities are frozen to known labels:

$$\psi_{i,s} = \begin{cases} 1, & \text{if } s = y_i \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The optimization process is then a propagation process on G, for which each probability at an iteration t is updated as the weighted average of its neighbors in the previous iteration:

$$\psi_{i,s}^{(t+1)} \propto \sum_{j \neq i} w_{i,j} \psi_{j,s}^{(t)} \tag{5}$$

and then normalized in order for individual probabilities to sum to 1.

The above-described method can be iterated until a maximum number of iterations t_{max} or an upper bound ϵ for the difference between iterations is reached. After convergence, an unknown label $\hat{y}_i$ (with $i > L$) is determined by

$$\hat{y}_i = arg \max_s \psi_{i,s}. \tag{6}$$

2.3 Time Series Classification

Regarding semi-supervised approaches for time series classification, these have been studied for quite a while [16,21,31,47], since the limitations on acquiring sufficient amounts of labeled data are present across many research areas. In this scenario, our main contribution is the study of a distance-based method for time series classification [1] using different methods for feature extraction, which will be discussed in the next section.

Under limited access to labeled data, graph-based approaches are widely employed. An earlier work employs a modified version of label propagation where at each step t the unlabeled instance with higher probability in step $t-1$ is considered as labeled [47] A more recent work uses a cluster-then-label approach [31] that aims at finding a minimum spanning tree for the similarity graph.

Approaches based on ensembles of classifiers are also popular in the field. HIVE-COTE [28] is an ensemble of classifiers for which the probability of a label is predicted in a hierarchical voting structure. Other methods in a similar direction are BOSS [37] and HESCA [19]. Despite HIVE-COTE being closer to being considered a state-of-the-art (SOTA) approach, the other two are still able to achieve better results on some datasets [28].

More recently, supervised deep learning methods have shown competitive results with more classical methods. The construction of a baseline [45] allowed for further improvements, like the use of long short-term memory layers [23] and ensembles of neural networks [15]. Semi-supervised approaches were also studied [17], finding that MixMatch is the most reliable method. This model, however, relies on augmenting unlabeled data during training.

As ensemble and deep learning methods demand a significant amount of computational resources, we focus our study on a simple classification method (Label Propagation) that can be efficiently implemented as a message-passing algorithm with time complexity $O(Nk)$, where k is the number of nearest neighbors for each node in the similarity graph.

In this context of more limited computational resources, the use of networks trained in a dataset that is not related to the task at hand, called transfer learning [40], is a recurrent approach. The removal of classification layers in order to use previous layers to extract features that could be more suitable to the problem and further application of a classical machine learning method (like Label Propagation) can overcome limitations such as the time required to train a deep neural network and also show that such models learn meaningful representations of data.

3 Proposed Approach

In this work, we propose a graph-based semi-supervised time series classification approach. First, we convert a series to an image by using three different methods, described in Sect. 3.1. For each generated image, we perform a feature extraction using two different neural networks, which are described in Sect. 3.2. In the next step, a graph with the entire dataset is built, and a Label Propagation is performed for the semi-supervised classification task, as described in Sect. 2.

3.1 Image-Based Time Series Description

Different approaches were proposed for time series feature extraction over the years. Shape-based [27], Symbol-based [26,38], and Kernel-based [9] are a few examples of time series feature extractors. Each feature can give a new perspective for a machine learning algorithm about a dataset. In this way, finding a good representation of data that can generalize information in a dataset and generate improved results in machine learning problems is a challenging task. An approach for time series representation is using images to extract features [11,43]. Different methods can generate an image using specific characteristics, such as a time series trajectory. These methods can be powerful tools for time series feature extraction if used along with an image feature extractor which is able to consider color, patterns, and texture information in the image. This step can be done using deep learning networks pre-trained through transfer learning, as it was done in this work. Below we describe three methods for imaging a time series.

Gramian Angular Fields (GAF) [43]: This method generates an image based on the polar coordinates of a time series. Given an time series $X = \{x_1, x_2, ...x_N\}$, a new series $\tilde{X}$ is obtained from the polar coordinates of the normalized series X, in range $[-1, 1]$. Considering the angular perspective, $\tilde{X}$

generates a Gramian Angular Field Matrix, with $N \times N$ dimension, described in Eq. 7.

$$GAF = \tilde{X}'\tilde{X} - \sqrt{I - \tilde{X}^2}'\sqrt{I - \tilde{X}^2} \tag{7}$$

where I is a unit line vector. Each point $GAF_{i,j}$ is the trigonometric sum of the intervals between these points. In this way, a Gramian Angular Field represents a temporal correlation between different time intervals in a time series.

Markov Transition Fields (MTF) [43]: This method images time series based on Markov Matrixes. Given a time series $X = \{x_1, x_2, ...x_N\}$, its Q quantile bins are identified, where $q_j(j\epsilon[1, Q])$ are points that divide a sorted time series into subsets with the same dimension. Given the quantile bins, the Markov Matrix M, with $Q \times Q$ dimension, is defined, as in the Eq. 8:

$$M = \begin{bmatrix} P(x_t \epsilon q_1 | x_{t-1} \epsilon q_1) & ... & P(x_t \epsilon q_1 | x_{t-1} \epsilon q_Q) \\ P(x_t \epsilon q_2 | x_{t-1} \epsilon q_1) & ... & P(x_t \epsilon q_2 | x_{t-1} \epsilon q_Q) \\ ... & ... & ... \\ P(x_t \epsilon q_Q | x_{t-1} \epsilon q_1) & ... & P(x_t \epsilon q_Q | x_{t-1} \epsilon q_Q) \end{bmatrix} \tag{8}$$

where $M_{i,j}$ describes the probability of x_t being in the quantile bin q_i given that x_{t-1} is in the quantile bin q_j. It removes the temporal correlation in the matrix [44]. In this way, a Markov Transition Field, with $N \times N$ dimension, is constructed according to the Eq. 9:

$$MTF = \begin{bmatrix} x_1 \epsilon q_i, x_1 \epsilon q_j & ... & x_1 \epsilon q_i, x_n \epsilon q_j \\ x_2 \epsilon q_i, x_1 \epsilon q_j & ... & x_2 \epsilon q_i, x_n \epsilon q_j \\ ... & ... & ... \\ x_n \epsilon q_i, x_1 \epsilon q_j & ... & x_n \epsilon q_i, x_n \epsilon q_j \end{bmatrix} \tag{9}$$

where $MTF_{i,j}$ is the probability of transition from a quantile bin q_i to a quantile bin q_j. Formally, $MTF_{i,j||i-j|=k}$ denotes the probability of transition between points with a time interval k.

Recurrence Plots (RP) [11]: A Recurrence Plot is a binary representation from a time series that gives information about the temporal correlation of a series. Given a time series $X = \{x_1, x_2, ...x_N\}$, it is obtained its trajectory $\vec{X} = \{\vec{x_i}, \vec{x_{i+\tau}}, ..., \vec{x_{i+(m-1)\tau}}\}$, $\forall i \epsilon \{1, 2, ..., N - (m-1)\tau\}$, where τ is the temporal delay and m is the trajectory dimension. A point $RP_{i,j}$ is set if the Euclidean distance between $\vec{x}(i)$ and $\vec{x}(j)$ is less or equal than a threshold. Formally, let $K = N-(m-1)\tau$, a Recurrence Plot matrix, with $K \times K$ dimension, is described in Eq. 10:

$$R_{i,j} = \begin{cases} 1, & \text{if } ||\vec{x}(i) - \vec{x}(j)|| \leq \epsilon \\ 0, & \text{else,} \end{cases} \tag{10}$$

$$\forall i, j \epsilon \{1, 2, ..., K\}$$

3.2 Deep Learning Methods as Image Feature Extractors

Images have lots of encoded information such as patterns, colors, textures, and shapes, among others. The use of previously trained neural networks via transfer learning, for image feature extraction, has shown great potential in this field due to its great generalization ability and strong feature learning [29]. In this work, we used a CNN ResNet-152 and a huge-sized Vision Transformers for feature extraction, both pre-trained with the ImageNet dataset [10] by using transfer learning techniques. ResNet architectures are widely used for image feature extraction [13,30,32] and have demonstrated excellent results in different scopes. On the other hand, although Vision Transformers have not been extensively tested for feature extraction, it has achieved competitive results in different tasks, such as image classification, making it a promising approach for this task. A brief explanation of these models is described below.

CNN ResNet-152 [18]. A Residual Network (ResNet) is a family of Convolutional Neural Networks (CNN) that was proposed for image recognition. This network can have up to 152 layers and overcomes the "vanishing gradient" problem by using skip connections on plain networks. A skip connection, also called identity shortcut, allows a more effective training process by enabling the flow of information between layers and avoiding the loss of gradient information during backpropagation. This facilitates the training of deeper neural networks. A skip connection involves taking multiple convolutional layers and skipping them, using activation functions of the previous layer. When the network is retrained, all of these layers, known as residual parts, are used for exploring more of the feature space, improving the accuracy of the model. For feature extraction, the last fully connected layer is used as output.

Vision Transformers (ViT) [25]. Vision Transformers is a model proposed for image recognition that achieved state-of-the-art in tasks such as object detection and image classification. The model uses a self-attention mechanism to enable the model to focus on the most important parts of the input by weighting and processing these parts simultaneously. The input image is represented as a sequence of patches that are flattened and fed into a transformers architecture. The transformers architecture consists of an encoder, that processes the input tokens and generates a sequence of hidden states, which feed a decoder, that produces a sequence of output tokens. Both the encoder and decoder are composed of multiple layers of multi-head self-attention mechanisms and feed-forward neural networks, but decoders use masked self-attention, so the model attends only to the previously generated tokens during decoding. For feature extraction, we used the pooler output.

4 Evaluation

This section outlines the experimental protocol for semi-supervised time series classification and the datasets used for this task. Our strategy can be summarized in four steps: first, we extract features from time series data; then, we split the

data and perform semi-supervised learning (SSL); finally, we evaluate the results of our approach.

4.1 Datasets

Below are described the four univariate time series datasets, from the UCR Archive [8], used in the experiments. The datasets were chosen based on size, the number of classes, and domain variability. This assures that our model is versatile, robust, and can be applied to a wide range of real-world problems. All datasets were z-normalized.

- **CBF** [36][1] This is a simulated dataset, where the data from each class consists of a standard normal noise plus an offset term that varies for each class. This dataset has 930 elements of size 128, divided into 3 classes.
- **ECG5000** [7][1] This dataset is derived from a 20-hour-long ECG called BIDMC Congestive Heart Failure Database (chfdb), available on Physionet. The used record is "chf07" and the patient has congestive heart failure. Data underwent pre-processing, extracting each heartbeat, and interpolating for uniform length. After pre-processing, 5000 heartbeats of length 140 were randomly selected, composing this dataset, and the heartbeats were automatically annotated and divided into five classes.
- **Yoga** [47][1] This dataset is composed of 3300 time series, of size 426, generated from videos of actors transitioning between yoga poses. Each image of the video was converted to a time series considering the distances between the actor's contour and the image center. The problem is distinguishing the male and the female actors.
- **Electric Devices** [2][1] This dataset has 16637 time series of length 96, divided into 7 classes. Data were obtained by measuring daily the power consumption of different devices, which are washing machines, ovens, dishwashers, kettles, immersion heaters, cold groups (fridge, freezer), and screen groups (computer, television), from 187 households. The problem consists in distinguishing each device by its daily measurements.

4.2 Experimental Protocol

The process of time series feature extraction was conducted using Python 3.10 and all the datasets were imaged using the methods described in Sect. 3.1, while we maintained all the proposed parameters on pyts library [14]. Images were plotted and saved using the matplotlib library [20]. For the GAF and MTF methods, we set the colormap parameter to *'rainbow'* and for the RP method, we used a *'binary'* colormap. The neural network architectures outlined in Sect. 3.2 were, then, applied to all generated images. Subsequently, we compute the Euclidean distance among pairs of instances in order to construct the similarity matrix for

[1] Downloaded from https://timeseriesclassification.com/dataset.php.

label propagation. For the inference step we set $t_{max} = 10^3$ and $\epsilon = 10^{-3}$.

To evaluate our methods, we split each dataset D in a labeled set D_l and an unlabeled one D_u, ensuring that the second always contains a representative of each class. This procedure is conducted for different sizes of D_u. All our experiments on classification are averaged over equally sized splits of D, which were ten in our setup.

The effect of the number of neighbors on classification results is also evaluated by varying k under a fixed size for D_u and then averaging over splits.

As classification metrics we employ the accuracy and adjusted mutual information, the second being a clustering metric, that, together with the first, helps in visualizing the relationship between clustering and classification in a SSL setup [3].

Comparison with previous works is done using supervised classification methods as baselines, particularly the ones in [23,28], which aggregate the accuracy for several methods and datasets. We then compare our accuracy results with the two best methods on each dataset.

For the extracted features, we conduct a statistical analysis in terms of the distribution of pairwise distances and similarities, and their coefficient of variation, defined as the standard deviation divided by the mean. This is done in order to have a better understanding of how different methods for feature extraction compare with each other.

5 Results and Discussion

We first discuss our empirical evaluation of different methods for feature extraction in semi-supervised classification and then proceed to our statistical analysis of pairwise distances to elucidate some of the differences among said methods.

5.1 Classification Results

Our results show the proposed methods for feature extraction are suitable for the task of semi-supervised classification using label propagation (Figs. 1, 2, 3 and 4). It is also noteworthy that accuracy and AMI carry a very intimate relationship in all of the studied datasets, showing clustering and classification are indeed connected.

The Electric Devices (Fig. 1) dataset presents some of our most interesting results. In this case, supervised baselines are trained with a high rate of labeled data ($r_l \approx 0.54$) and one can clearly see that with $r_l = 0.2$ some of the proposed methods are not far away of their accuracy. It is also noteworthy that using GAF and RP achieves higher accuracies with $r_l > 0.06$ than methods using the combination LSTM+FCN [23] in the standard splits.

Another interesting feature of this dataset is that, different from the others, using raw data provides the worst results. To our knowledge, this dataset does not have a Gaussian distribution. However, the discrepancy from the other

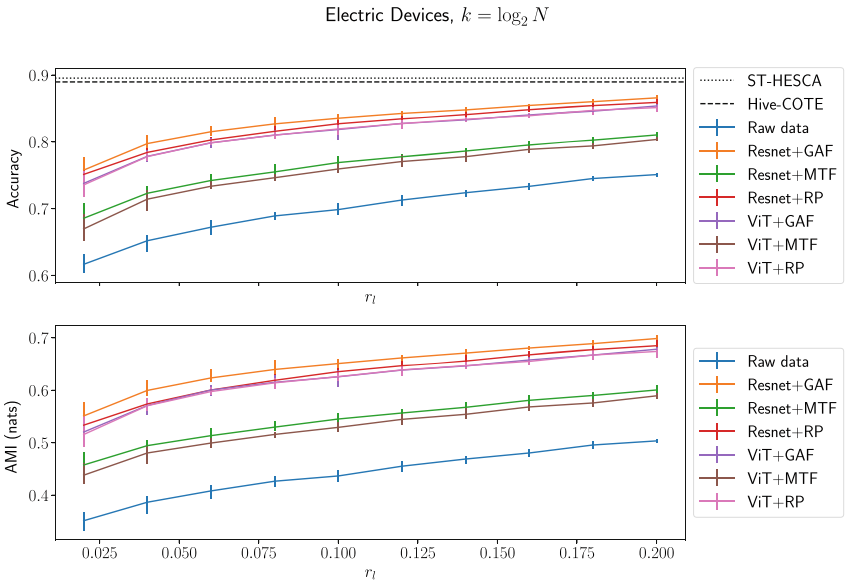

Fig. 1. Results for the ElectricDevices dataset as a function of r_l. Supervised baselines were obtained from 8926 labeled points, corresponding to $r_l \approx 0.54$.

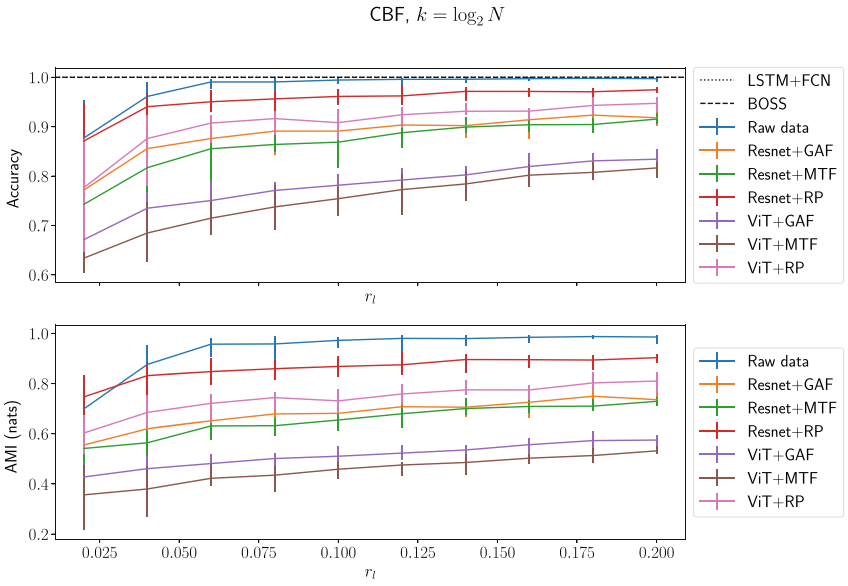

Fig. 2. Results for the CBF dataset as functions of the number of nearest neighbors k (up) and the rate of labeled data r_l (down). Supervised baselines were obtained with initially 30 labeled points, corresponding to $r_l \approx 0.03$.

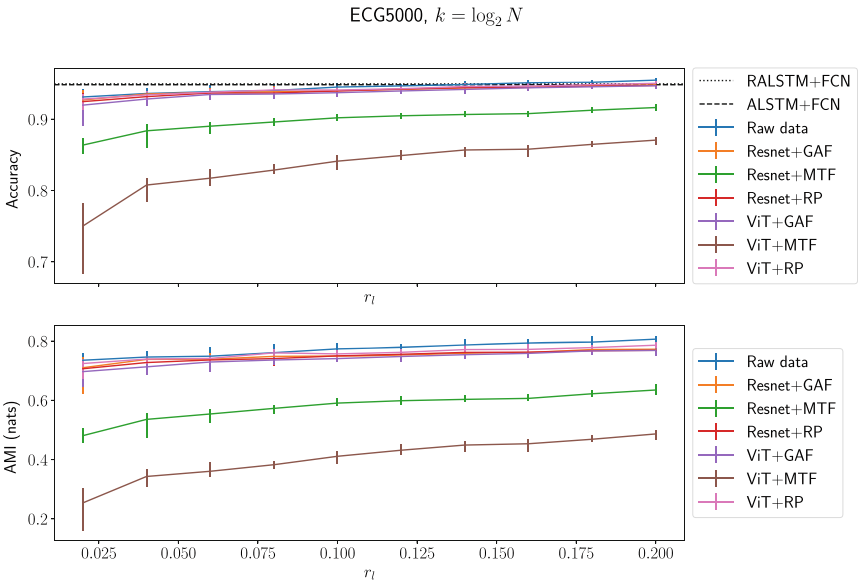

Fig. 3. Results for the ECG5000 dataset as a function of r_l. Supervised baselines were obtained with initially 500 labeled points, corresponding to $r_l = 0.1$.

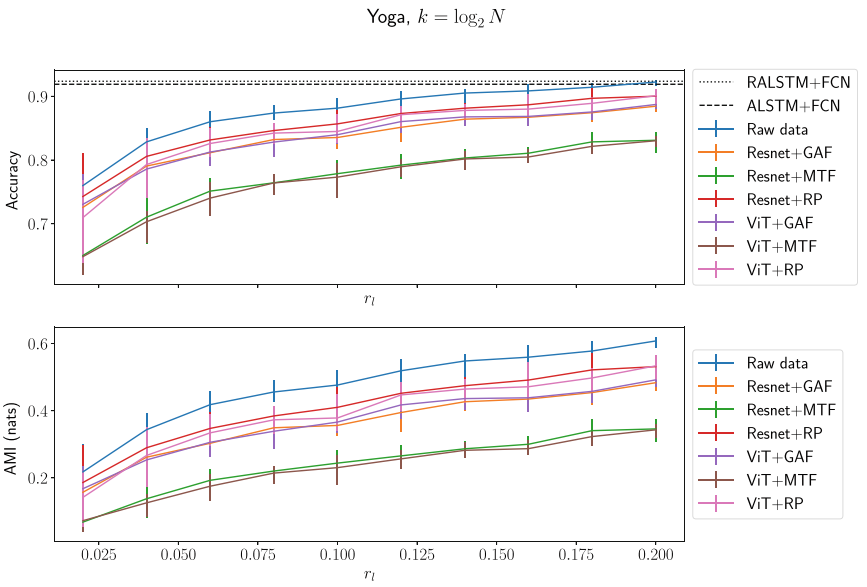

Fig. 4. Results for the Yoga dataset as a function of r_l. Supervised baselines were obtained with initially 300 labeled points, corresponding to $r_l \approx 0.09$.

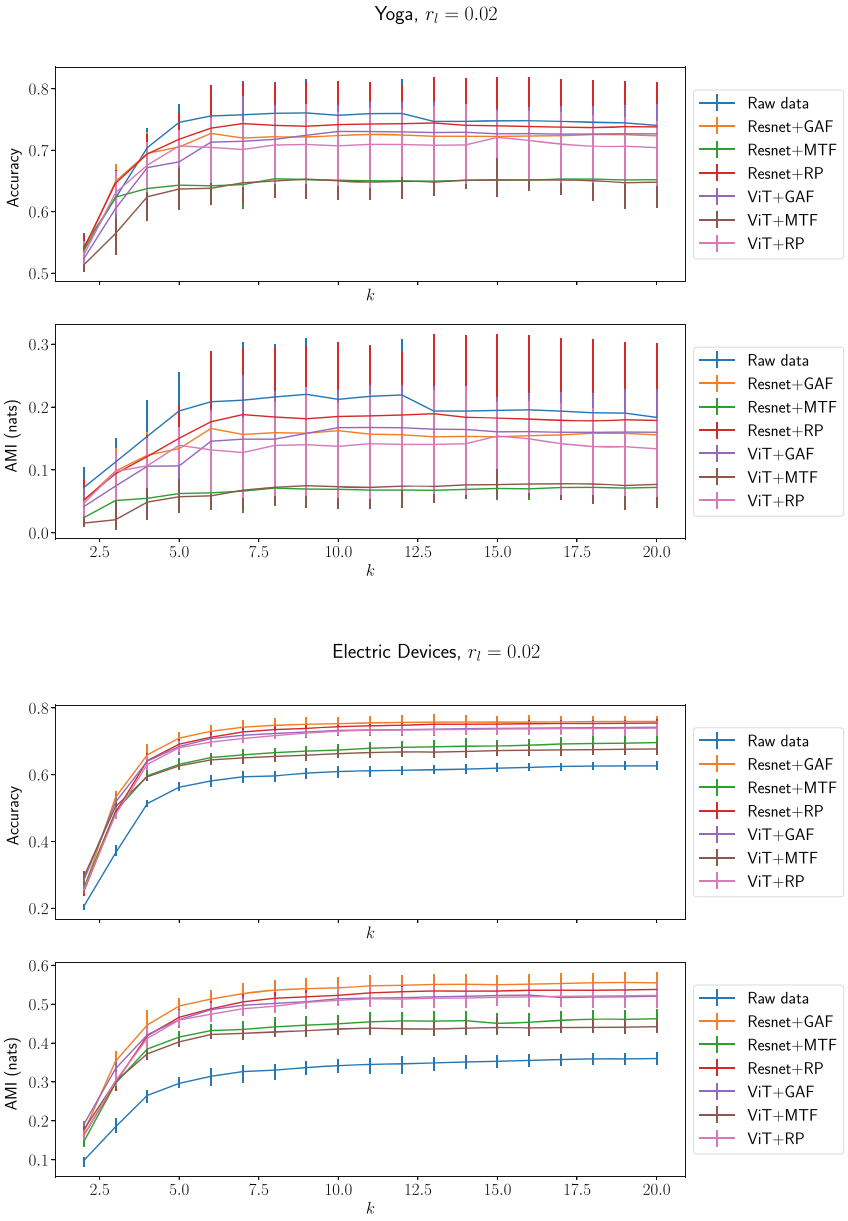

Fig. 5. Classification results as a function of k for the Yoga (up) and Electric Devices (down) datasets.

studied sets may indicate a line of work for which our methods may provide a significant improvement.

One also sees that, with the exception of Electric Devices, the usage of raw data dominates our benchmarks in comparison with methods that use features extracted via neural networks. This could be related to CBF, ECG5000, and Yoga having some form of Gaussian property, either explicitly, as is the case of CBF, or intrinsically. This second hypothesis, however, demands a deeper evaluation to be better understood.

Drawing our attention to comparisons with previous works, on CBF (Fig. 2), previous works using neural networks [23] and also BOSS [38] trained with $r_l \approx 0.03$ achieve correctness over the whole dataset, while our methods fall shortly of that milestone. On ECG5000 we observe our methods are only marginally less accurate than LSTM+FCN ones [23]. Yoga is the hardest dataset, with our methods demanding $r_l > 0.2$ in order to compete with SOTA accuracy.

We also observe that methods that rely on MTF for imaging of time series tend to be less accurate than their counterparts using GAF and RP. This is also a strange behavior that we are not yet able to understand.

Regarding the role of graph construction on classification results, we show results for Yoga and Electric Devices (Fig. 5) under $r_l = 0.02$ and variable k. As noted in earlier works [3,39], there is evidence of a *plateau* in studied metrics once a sufficient amount of nearest neighbors is reached.

Fig. 5 also shows us that is the proposed feature extraction approach that renders Electric Devices its performance and not an anomaly due to graph construction. As different values of k do not alter the relative classification performance among evaluated methods, we are able to confirm that our imaging of time series can indeed provide a representation of data that is more discriminative among classes.

The choice for k in this method, however, defines also the computational complexity. Therefore, one should choose k in a way that the method is fast and accurate. From our experiments, the initial choice of $k = \log_2 N$ seems a good one to attain these conditions.

5.2 Statistical Analysis of Pairwise Distances and Similarities

We now turn to analyzing the distribution of pairwise distances and similarities. This is done only on CBF, ECG5000, and Yoga due to the large size of the Electric Devices dataset.

Fig. 6 shows the kernel density estimation of the distribution of distances, obtained using the Seaborn library [46]. One then can easily see that features extracted using ViT networks produce pairwise distances that are smaller than other evaluated methods.

Another distributional aspect that draws our attention is that the distribution of pairwise distances of the most accurate methods tends to not resemble a Gaussian. In fact, some of those distributions even show a double-peaked distribution, being indicative of a separation between intraclass and outer class distances. This is the case for using raw data in ECG5000 and Resnet+RP in CBF.

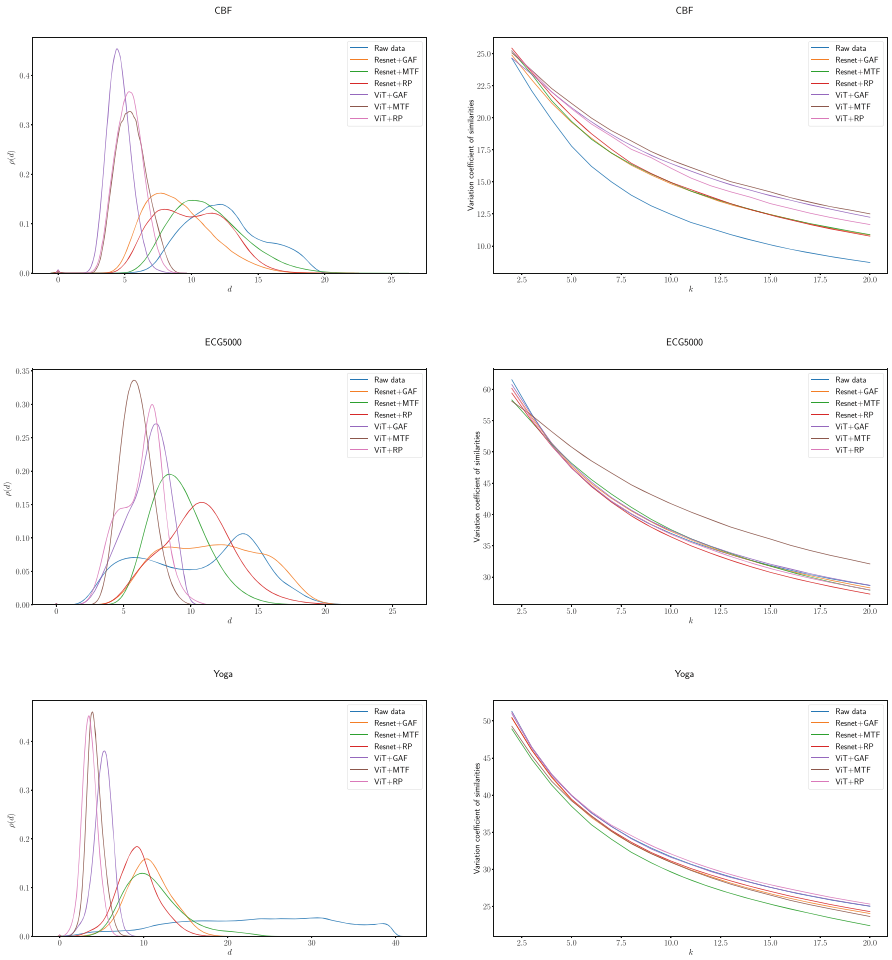

Fig. 6. Kernel density estimation of pairwise distances (left) and coefficient of variation of pairwise similarities as a function of k (right) for CBF, ECG5000, and Yoga.

Results in Fig. 6 also outline that the combination of imaging with feature extraction from neural networks tends to smooth the distribution of distances. ResNet and ViT, however, do such in a different manner, with the second producing smaller distances.

When we turn to the analysis of the coefficient of variation of similarities (Fig. 6) we see that, on CBF, the distribution for raw data has the lowest variation. Together with our previous analysis of Fig. 6, we then conclude that similarities may reflect intraclass proximity in this case.

However, especially on the Yoga dataset, similarities calculated from MTF imaging have a lower variation. This dataset and ECG5000 are also the ones where MTF is clearly the worst imaging method. As their kernel densities (Fig. 6)

show a single-peaked distribution, we presume pairwise distances of these methods suffer from an indistinguishability phenomenon.

6 Conclusion

We have studied imaging of time series followed by feature extraction from previously trained deep neural networks applied to semi-supervised classification.

Our results indicate that the proposed approach is useful in the problem of time series classification, showing competitive results on most datasets.

The conducted empirical evaluation, however, leaves lots of questions to be answered. Classification results on CBF, ECG5000, and Yoga are somewhat similar, where the usage of raw data instead of imaging is the most effective approach. On ElectricDevices this picture changes drastically and imaging with GAF and RP, followed by MTF, provides the best results. Understanding the causes of this phenomenon is our main research direction moving forward.

It is also noteworthy that imaging methods seem to be way more impactful on classification results than neural network architectures. In this regard, MTF is clearly the worst approach when compared to GAF and RP.

The statistical analysis of pairwise distances and similarities also helped us with some of our classification results and revealed interesting properties of features extracted from neural networks, like the fact that Vision Transformers produce more homogeneous and smaller distances. Studying such aspects also seems a fruitful research direction we aim to purchase in future work.

Acknowledgements. The authors are grateful to São Paulo Research Foundation - FAPESP (grants #2016/05669-4, #2022/01359-1 and #2018/15597-6), Brazilian National Council for Scientific and Technological Development - CNPq (grants #309439/2020-5 and #422667/2021-8) for financial support. This study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

We thank Prof. Dr. Denis Salvadeo, from DEMAC, for some insightful discussions and help with computational infrastructure.

References

1. Abanda, A., Mori, U., Lozano, J.A.: A review on distance based time series classification. Data Min. Knowl. Disc. **33**(2), 378–412 (2019)
2. Bagnall, A., Davis, L., Hills, J., Lines, J.: Transformation based ensembles for time series classification. In: Proceedings of 12th SDM, April 2012. https://doi.org/10.1137/1.9781611972825.27
3. Bergamim, E., Breve, F.: On tuning a mean-field model for semi-supervised classification. J. Stat. Mech: Theory Exp. **2022**(5), 053402 (2022)
4. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: a holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

5. Breve, F., Zhao, L., Quiles, M., Pedrycz, W., Liu, J.: Particle competition and cooperation in networks for semi-supervised learning. IEEE Trans. Knowl. Data Eng. **24**(9), 1686–1698 (2011)
6. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. Adaptive computation and machine learning series (2006)
7. Chen, Y., Hao, Y., Rakthanmanon, T., Zakaria, J., Hu, B., Keogh, E.: A general framework for never-ending learning from time series streams. Data Min. Knowl. Disc. **29**(6), 1622–1664 (2014). https://doi.org/10.1007/s10618-014-0388-4
8. Dau, H.A., et al.: The UCR time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/
9. Dempster, A., Petitjean, F., Webb, G.I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. **34**(5), 1454–1495 (2020). https://doi.org/10.1007/s10618-020-00701-z
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
11. Eckmann, J.P., Kamphorst, S., Ruelle, D.: Recurrence plots of dynamical systems. Europhys. Lett. (EPL). **4**, 973–977 (1987). https://doi.org/10.1209/0295-5075/4/9/004
12. Falcon, A., D'Agostino, G., Lanz, O., Brajnik, G., Tasso, C., Serra, G.: Neural turing machines for the remaining useful life estimation problem. Comput. Indus. **143**, 103762 (2022). https://doi.org/10.1016/j.compind.2022.103762
13. Fan, L., Zhang, T., Du, W.: Optical-flow-based framework to boost video object detection performance with object enhancement. Expert Syst. Appl. **170**, 114544 (2021). https://doi.org/10.1016/j.eswa.2020.114544
14. Faouzi, J., Janati, H.: PYTS: a python package for time series classification. J. Mach. Learn. Res. **21**(46), 1–6 (2020). http://jmlr.org/papers/v21/19-763.html
15. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep neural network ensembles for time series classification. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2019)
16. González, M., Bergmeir, C., Triguero, I., Rodríguez, Y., Benítez, J.M.: Self-labeling techniques for semi-supervised time series classification: an empirical study. Knowl. Inf. Syst. **55**, 493–528 (2018)
17. Goschenhofer, J.: Deep semi-supervised learning for time-series classification. In: Wani, M.A., Palade , V. (eds.) Deep Learning Applications, Volume 4. Advances in Intelligent Systems and Computing, vol. 1434, pp. 361–384. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-6153-3_15
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
19. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by Shapelet transformation. Data Min. Knowl. Disc. **28**, 851–881 (2014)
20. Hunter, J.D.: Matplotlib: a 2d graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55
21. Jawed, S., Grabocka, J., Schmidt-Thieme, L.: Self-supervised learning for semi-supervised time series classification. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12084, pp. 499–511. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47426-3_39

22. Jebara, T., Wang, J., Chang, S.F.: Graph construction and b-matching for semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 441–448 (2009)
23. Karim, F., Majumdar, S., Darabi, H., Chen, S.: LSTM fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2017)
24. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
25. Kolesnikov, A., et al.: An image is worth 16 × 16 words: Transformers for image recognition at scale (2021)
26. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**, 107–144 (2007). https://doi.org/10.1007/s10618-007-0064-z
27. Lines, J., Davis, L.M., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 289–297. KDD 2012, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2339530.2339579
28. Lines, J., Taylor, S., Bagnall, A.: Time series classification with hive-cote: the hierarchical vote collective of transformation-based ensembles. ACM Trans. Knowl. Disc. Data **12**(5) (2018)
29. Liu, Y., Pu, H., Sun, D.W.: Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends Food Sci. Technol. **113**, 193–204 (2021). https://doi.org/10.1016/j.tifs.2021.04.042
30. Mahajan, A., Chaudhary, S.: Categorical image classification based on representational deep network (ResNet). In: 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 327–330 (2019). https://doi.org/10.1109/ICECA.2019.8822133
31. Marussy, K., Buza, K.: SUCCESS: a new approach for semi-supervised classification of time-series. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013. LNCS (LNAI), vol. 7894, pp. 437–447. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38658-9_39
32. Nidhyananthan, S.S., Shebiah, R.N., Kumari, B.V., Gopalakrishnan, K.: Chapter 15 - deep learning for accident avoidance in a hostile driving environment. In: Zhang, Y.D., Sangaiah, A.K. (eds.) Cognitive Systems and Signal Processing in Image Processing, pp. 337–357. Cognitive Data Science in Sustainable Computing, Academic Press (2022). https://doi.org/10.1016/B978-0-12-824410-4.00002-7
33. Pincus, S., Kalman, R.: Irregularity, volatility, risk, and financial market time series. In: Proceedings of the National Academy of Sciences of the USA, vol. 101, pp. 13709–13714 (2004)
34. Pino, F.A.: Sazonalidade na agricultura. Revista De Economia Agrícola (Printed) **61**, 63–93 (2014)
35. Saeed, A., Ozcelebi, T., Lukkien, J.: Multi-task self-supervised learning for human activity detection. CoRR abs/1907.11879 (2019)
36. Saito, N.: Local feature extraction and its applications using a library of bases. Yale University (1994)
37. Schäfer, P.: The boss is concerned with time series classification in the presence of noise. Data Min. Knowl. Disc. **29**, 1505–1530 (2015)

38. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Min. Knowl. Disc. **29**(6), 1505–1530 (2014). https://doi.org/10.1007/s10618-014-0377-7
39. de Sousa, C.A.R., Rezende, S.O., Batista, G.E.A.P.A.: Influence of graph construction on semi-supervised learning. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8190, pp. 160–175. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_11
40. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11141, pp. 270–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01424-7_27
41. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2020)
42. Volna, E., Kotyrba, M., Habiballa, H.: ECG prediction based on classification via neural networks and linguistic fuzzy logic forecaster. Sci. World J. **2015**, 205749 (2015). https://doi.org/10.1155/2015/205749
43. Wang, Z., Oates, T.: Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: Workshops at the Twenty-ninth AAAI Conference on Artificial Intelligence (2015)
44. Wang, Z., Oates, T.: Imaging time-series to improve classification and imputation. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 3939–3945. AAAI Press (2015)
45. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1578–1585. IEEE (2017)
46. Waskom, M.L.: Seaborn: statistical data visualization. J. Open Sourc. Softw. **6**(60), 3021 (2021)
47. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 748–753 (2006)
48. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems, vol. 16 (2003)
49. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)