Evaluate Pseudo Labeling and CNN for multi-variate time series classification in low-data regimes

Dino Ienco*[1], Davi Pereira-Santos[2], and André C.P.L.F. de Carvalho[2]

[1] INRAE, UMR TETIS,
Univ. Montpellier, Montpellier, France
dino.ienco@inrae.fr
[2] ICMC, Sao Carlos, Brazil
{davips,andre}@icmc.usp.br

Abstract. Nowadays, huge amount of data are being produced by a large and diverse family of sensors (e.g., remote sensors, biochemical sensors, wearable devices). These sensors typically measure multiple variables over time, resulting in data streams that can be profitably organized as multivariate time-series. In practical scenarios, the speed at which such information is collected often makes the data labeling a difficult task. This results in a low-data regime scenario where only a small set of labeled samples is available and standard supervised learning algorithms cannot be employed.

To cope with the task of multi-variate time series classification in low-data regime scenarios, here, we propose a framework that combines convolutional neural networks (CNNs) with self-training (pseudo labeling) in a transductive setting (test data are already available at training time). Our framework, named $ResNet^{IPL}$, wraps a CNN based classifier into an iterative procedure that, at each step, enlarges the training set with new samples and their associated pseudo labels. An experimental evaluation on several benchmarks, coming from different domains, has demonstrated the value of the proposed approach and, more generally, the ability of the deep learning approaches to effectively deal with scenarios characterized by low-data regimes.

1 Introduction

A vast amount of information is generated by a widespread and diverse family of sensors like remote sensors, biochemical sensors and wearable devices. They typically measure multiple variables over time, resulting in data streams that can be profitably organized as multivariate time-series. Due to the ubiquitous nature of multivariate time-series, conceiving classification methods especially tailored for such kind of data is crucial [16]. In a more realistic but challenging scenario, only a limited set of samples, among the available data, is associated with label

* Corresponding author

information resulting in a low-data regime scenario that requires effective semi-supervised learning methods [2].

Regarding the semi-supervised classification of time series data, [9] introduces an approach that firstly uses hierarchical clustering to group labeled and unlabeled data, then propagates label information inside each cluster and, finally, employs a one nearest neighbors (1NN) classifier with dynamic time warping (DTW) to perform classification. A similar approach is proposed in [5] where clustering and a 1NN classifier are combined to perform classification of univariate time series when only a few labeled data are available. [19] recently introduces a multivariate time series classification based on neural attentional prototype network to train the feature representation based on their distance to class prototypes considering low-data regimes. The proposed deep learning based method works considering both fully supervised and semi-supervised settings.

Another family of semi-supervised methods for time series classification is based on self-training (or self-labeling) approaches, whose goal is to enlarge the original labeled set selecting the unlabeled samples with the most confident predictions [14]. In conjunction with the self-training framework, the 1NN classifier is typically used as the base learner, as it has been effective for time series classification tasks [2].

Recently, [2] evaluated several machine learning based approaches to deal with the semi-supervised classification of univariate time series. The approach proposed by the authors is to couple standard classifiers with pseudo labeling and self-training strategies. The results underline that, among all considered classifiers, 1NN still exhibits superior performance as base learner. Unfortunately, the study is limited to univariate time series while multivariate time-series are becoming more and more predominant nowadays. Additionally, it totally ignores the recent advent of deep learning (DL) approaches in the time series community [4, 19]. Indeed, despite the recent findings reported in [4] where a DL strategy (residual-based convolutional neural networks) exhibits superior performance in standard fully supervised classification tasks, no discussion is reported about the appropriateness of such approaches when only few labeled time-series data are available to learn a classification model. This fact indicates that, considering the classification of time series data in a low-data regime scenario, the use of deep learning approaches is still an under explored field of research.

We propose, here, a framework that combines deep learning with pseudo labeling in a *transductive setting*, i.e., when test data are already available during model training. Our approach is motivated by the lack of studies that explore the application of deep learning to low-data regime scenarios, with a particular emphasis on multivariate time series classification tasks. The proposed framework, referred to as $ResNet^{IPL}$ (ResNet with Incremental Pseudo Labeling), wraps the deep learning based classifier into an iterative procedure that, at each step, enlarges the training set with new samples and their associated pseudo labels. The sample selection stage leverages the classifier prediction on unlabeled data and chooses those samples that minimize the relative entropy associated to the model output distribution. To assess the performance of the proposed framework as well as its generality, we conduct an extensive experimental evaluation on several multivariate time-series benchmarks coming from different domains.

The rest of the manuscript is organized as follows: the $ResNet^{IPL}$ framework is introduced in Section 2, experimental settings as well as experimental evaluation are described in Section 3 while Section 4 draws conclusions and possible follow-ups.

2 Methodology

In this section, we describe our proposed incremental pseudo labeling procedure for the classification of multi-variate time series data considering a low-data regime scenario.

The general procedure is depicted in Algorithm 1. Due to the fact that we are considering a transductive scenario, test samples are available at training time. The procedure takes as input the set of training samples (X_{train}) with the associated labels (Y_{train}), the test samples X_{test}, the number of iterations of the incremental procedure (T), and the number of samples per class added at each iteration to the training set (k). The output of the procedure is a multi-variate time series classification model that is trained with both original and pseudo labeled samples. At the beginning, the classification model ($Classifier$) is initialized and then trained on the original labeled samples X_{train}, Y_{train} (Line 1-2). Then, the incremental process starts (Line 4-13). At each iteration, the classification model is applied on the current test data (X_{test}) and the class distribution for each test sample (derived by the softmax layer of the classification model) is obtained (Line 5). The class distribution, the unlabeled data X_{test} and the k parameter are employed by the sample selection procedure. This procedure extracts, for each class, k reliable examples according to the class distribution previously outputted by the classification model. The set of selected samples and their associated pseudo labels are referred to as X_{sel} and *pseudolLabel*, respectively (Line 6). Successively, the training and testing sets are updated according to the results of the sample selection procedure (Line 7-9). Finally, the classification model is initialized again and trained on the new set of training samples that combines both original and pseudo label information (Line 10-11).

Regarding the general procedure depicted in Algorithm 1, two points must be defined in order to deploy such strategy: firstly, the choice of the classification model and, secondly, the implementation of the sample selection procedure.

Concerning the classification model, we base our choice on the findings reported in a recent literature survey [4]. Among several deep learning architectures for time series classification, the Residual Network $ResNet$ model proposed in [17] exhibits superior behavior. Due to this fact, we choose such architecture as classification model in our study. The network is composed of three residual blocks followed by a GAP (Global Average Pooling) layer and a final softmax classifier whose number of neurons is equal to the number of classes in a dataset. Each residual block is first composed of three convolutions whose output is added to the residual block's input and then fed to the next layer. The number of fil-

Algorithm 1 Incremental Pseudo Labeling procedure

Require: X_{train}, Y_{train}, X_{test}, T, k.
Ensure: $Classifier$.
$Classifier \leftarrow$ initModel()
$Classifier \leftarrow$ TrainModel(Classifier, X_{train}, Y_{train})
$i \leftarrow 0$
while i < T **do**
  classDistrib $\leftarrow$ Classify($classifier$, X_{test})
  X_{sel}, pseudoLabel $\leftarrow$ SampleSelection(X_{test}, classDistrib, k)
  $X_{train} \leftarrow X_{train} \cup X_{sel}$
  $Y_{train} \leftarrow Y_{train} \cup pseudoLabel$
  $X_{test} \leftarrow X_{test} - X_{sel}$
  $Classifier \leftarrow$ initModel()
  $Classifier \leftarrow$ TrainModel(Classifier, X_{train}, Y_{train})
  $i \leftarrow i + 1$
end while
return $Classifier$

ters for all convolutions is fixed to 64, with the ReLU activation function that is preceded by a batch normalization operation. In each residual block, the filter's length is set to 8, 5 and 3 respectively for the first, second and third convolution.

The second point involves the definition of a sample selection strategy. Such a strategy is mainly based on the analysis of the class distribution output by the classification model. More in detail, for each sample x_t we exploit the class distribution $pd(x_t)$. $pd(x_t)$ is the probability distribution over all possible classes that corresponds to the softmax output of the classification model regarding the sample x_t. Our strategy selects unlabeled samples on which the classifier has the highest confidence. To this purpose, we consider as surrogate of the confidence measure the entropy over the classifier output probability distribution. The entropy measure has already demonstrated its quality in pseudo labeling strategies to select valuable samples in the context of image analysis and semantic segmentation [10].

In our case, we adopt a normalized version of the entropy measure defined as follows:

$$H(x_t) = -\frac{\sum_{c \in C} pd_c(x_t) \times \log(pd_c(x_t))}{\log(|C|)} \tag{1}$$

where C is the set of possible classes and $pd_c(x_t)$ is the probability of sample x_t to belong to class $c \in C$. Samples with low entropy values correspond to time series on which the classifier has high confidence in its prediction. The $SampleSelection()$ procedure is summarized in Algorithm 2.

The procedure takes as input the set of test samples (X_{test}), the class distribution obtained by the classification model $ClassDistrib$ and the parameter k corresponding to the number of per-class samples to select. The output of the procedure is the set of the selected samples (X_{sel}) with their associated pseudo labels ($pseudolLabel$).

We can note that the set of selected samples (X_{sel}) with the associated pseudo labels ($pseudoLabel$) is obtained class by class (Line 3-8). For each class $c \in C$, we select the samples that the classifier judges to belong to that class (Line 4). Successively, the selected samples (X_c) are ranked in ascending order w.r.t. the entropy measure defined in Equation 1. Finally, the top K samples ($K = \{x_i \mid x_i \in X_c, 1 \leq i < k\}$) are added to the final set along with their corresponding pseudo labels. $[c]^k$ indicates a vector where the class value c is repeated k times.

Algorithm 2 SampleSelection(X_{test}, ClassDistrib, k)

Require: X_{test}, ClassDistrib, k.
Ensure: X_{sel}, pseudoLabel.
$X_{sel} \leftarrow \emptyset$
$pseudoLabel \leftarrow \emptyset$
for all $c \in C$ **do**
  $X_c \leftarrow \{x | x \in X_{test}, [\operatorname{argmax}_{v \in C} ClassDistrib_v(x)] = c\}$
  rank X_c in ascending order considering the entropy measure defined in Equation 1
  $K \leftarrow \{x_i \mid x_i \in X_c, 1 \leq i < k\}$
  $X_{sel} \leftarrow X_{sel} \cup K$
  $pseudoLabel \leftarrow pseudoLabel \cup [c]^k$
end for
return X_{sel}, $pseudoLabel$

3 Experimental Evaluation

In this section we assess the behavior of our framework considering five real world multivariate time series benchmarks. To evaluate the performance of our proposal, we compare $ResNet^{IPL}$ with several competing and baseline approaches.

3.1 Competitors and method ablations

For the comparative study, we consider the following competitors:

- A one nearest neighbors classifier (1NN) coupled with the DTW measure [3]. 1NN is a well recognized and widely adopted classifier in the time series classification domain [7, 2]. We name such competitor $1NN_{DTW}$.
- A graph-based semi-supervised learning approach since we are considering a transductive scenario. Among the different available methods, we adopted the CAMLP (Confidence-Aware Modulated Label Propagation) approach [18] as a representative one. Since CAMLP requires the construction of a K-nearest-neighbors graph to perform its propagation process, we chose to set K equals to 20, according to the study proposed in [11], and construct the K-nearest-neighbors graph leveraging, also in this case, the DTW similarity measure. We name this competitor as $GBSSL_{DTW}$.
- The $ResNet$ approach proposed in [4] without the incremental pseudo labeling strategy. This competitor can be seen as an ablation of the proposed framework. We name this baseline as $ResNet$.
- The recent TapNet approach [19] which introduces a multivariate time series classification with attentional prototypical deep neural network. Due to the transductive setting considered in our work, we adopt the semi-supervised version that exploits unlabelled data during training.

For the $ResNet$ [3] and TapNet models [4], we use their available implementations.

Furthermore, we couple $1NN_{DTW}$ and $GBSSL_{DTW}$ with the proposed incremental pseudo labeling strategy. These additional competitors are referred to as $1NN_{DTW}^{IPL}$ and $GBSSL_{DTW}^{IPL}$, respectively.

$ResNet$ is implemented via the Tensorflow 2 python library [5] while the implementation of the remain competitors is based on TSLEAN [13] and SCIKIT-learn [1] python libraries.

3.2 Data and Experimental Settings

The evaluation has been carried out by performing experiments on five benchmarks coming from disparate application domains and characterized by contrasted features in terms of number of samples, number of attributes (dimensions) and time length. All benchmarks, except *Dordogne* – which was obtained contacting the authors of [6], are available online.

Table 1: Benchmarks Characteristics

Dataset	# Samples	# Dims	Min/Max Length	Avg. Length	# Classes
Dordogne	9 919	6	23/23	23	7
GTZAN	600	33	128/128	128	6
HAR	10 299	9	128/128	128	6
JapVowel	640	12	7/29	15	9
SpeechCom	23 682	40	14/32	31	10

The characteristics of the five benchmarks are reported in Table 1. For each benchmark, we consider different amount of per-class labeled samples. The amount of per-class labeled samples ranges in the set {10,15,20,25}. This means that, for instance, considering the value 10, ten samples per class are randomly chosen and used to compose the initial training set, and the rest of them is considered as the test set. For all the methods that involve the incremental pseudo labeling procedure, 10 samples per class are moved, at each round (according to the strategy specified in Section 2) from the test set to the training set and associated to the pseudo labels estimated by the specific learning algorithm. Classification performances are assessed by F-Measure metric [12] considering the original test set. F-Measure is chosen as metric due to its ability to take into account possible class imbalance scenarios.

The obtained results are averaged over five different runs for each given method and benchmark, due to the non deterministic nature of the sample selection. Finally, the average value is reported.

[3] https://github.com/hfawaz/dl-4-tsc

[4] https://github.com/xuczhang/tapnet

[5] Code will be available upon acceptance

3.3 Quantitative results

Tables 2, 3, 4, 5, 6 depict the performance results, in terms of F-Measure (average and standard deviation), of the different competing approaches varying the amount of available label data. We can observe that $ResNet$ always outperforms the competing approaches ($1NN$ and $GBSSL$) considering all the five benchmarks as well as all the training size. Regarding non deep learning approaches ($1NN$ and $GBSSL$), we can note that no approach systematically outperforms the other. Despite $GBSSL$ is not largely adopted by the time series classification community, it exhibits comparable behavior w.r.t. the commonly adopted $1NN$ method. In addition, coupling such competitors with the incremental pseudo labeling framework always ameliorate the method performances no matter the training size. Regarding the $1NN$ approach behavior, this is in line with the experimental findings reported in [2] for univariate time series.

Table 2: F-Measure results over the Dordogne benchmark varying the amount of labelled examples per class. We report average and standard deviation. Bold and underlined text indicate best and second-best results, respectively.

	10	15	20	25
$1NN_{DTW}$	14.25 ± 1.58	14.99 ± 3.18	16.49 ± 2.43	16.05 ± 1.67
$GBSSL_{DTW}$	60.03 ± 1.58	61.82 ± 2.20	63.00 ± 2.30	64.24 ± 1.40
$1NN_{DTW}^{IPL}$	58.32 ± 2.07	60.81 ± 2.61	61.02 ± 1.76	61.71 ± 1.54
$GBSSL_{DTW}^{IPL}$	62.57 ± 2.50	63.71 ± 2.99	64.53 ± 2.46	65.24 ± 1.91
TapNet	60.97 ± 1.97	63.78 ± 2.52	65.33 ± 2.17	67.20 ± 1.17
$ResNet$	**64.70** ± 2.46	<u>66.47</u> ± 2.65	**68.94** ± 1.99	**70.35** ± 1.46
$ResNet^{IPL}$	<u>63.15</u> ± 2.31	**67.11** ± 2.78	<u>68.69</u> ± 2.64	<u>70.02</u> ± 1.35

Table 3: F-Measure results over the GTZAN benchmark varying the amount of labelled examples per class. We report average and standard deviation. Bold and underlined text indicate best and second-best results, respectively.

	10	15	20	25
$1NN_{DTW}$	16.59 ± 2.09	17.02 ± 1.66	15.47 ± 2.09	16.26 ± 1.59
$GBSSL_{DTW}$	47.77 ± 1.11	49.84 ± 2.26	50.48 ± 2.32	51.96 ± 1.72
$1NN_{DTW}^{IPL}$	62.86 ± 2.24	65.33 ± 2.10	67.23 ± 3.05	68.05 ± 2.68
$GBSSL_{DTW}^{IPL}$	63.58 ± 1.81	65.74 ± 2.15	67.05 ± 0.55	69.30 ± 1.24
TapNet	44.98 ± 1.08	45.83 ± 1.75	47.85 ± 1.09	48.40 ± 1.59
$ResNet$	<u>64.90</u> ± 1.48	<u>69.18</u> ± 3.06	<u>71.59</u> ± 3.72	<u>72.90</u> ± 2.74
$ResNet^{IPL}$	**67.25** ± 1.98	**72.83** ± 1.63	**74.18** ± 0.96	**76.65** ± 2.20

Regarding the proposed framework, we observe that incremental pseudo labeling achieves better performances than its counterpart without pseudo labeling

Table 4: F-Measure results over the HAR varying the amount of labelled examples per class. We report average and standard deviation. Bold and underlined text indicate best and second-best results, respectively.

	10	15	20	25
$1NN_{DTW}$	16.82 ± 2.29	16.92 ± 3.11	15.59 ± 2.40	17.25 ± 3.16
$GBSSL_{DTW}$	60.29 ± 4.01	65.43 ± 3.64	69.13 ± 1.98	70.48 ± 1.44
$1NN_{DTW}^{IPL}$	78.90 ± 2.12	81.34 ± 2.11	83.07 ± 1.79	84.45 ± 1.39
$GBSSL_{DTW}^{IPL}$	60.61 ± 3.86	65.94 ± 3.10	69.67 ± 1.61	70.93 ± 0.96
TapNet	65.41 ± 2.24	67.49 ± 2.15	69.53 ± 1.93	70.41 ± 1.78
$ResNet$	<u>87.87</u> ± 2.71	<u>89.66</u> ± 2.51	**90.96** ± 0.71	**90.26** ± 0.90
$ResNet^{IPL}$	**88.28** ± 2.13	**90.15** ± 1.23	<u>89.10</u> ± 2.64	<u>88.98</u> ± 2.14

Table 5: F-Measure results over the JapVowel benchmark varying the amount of labelled examples per class. We report average and standard deviation. Bold and underlined text indicate best and second-best results, respectively.

	10	15	20	25
$1NN_{DTW}$	6.88 ± 1.57	8.68 ± 3.37	7.73 ± 1.65	7.60 ± 2.65
$GBSSL_{DTW}$	88.07 ± 1.14	89.14 ± 0.58	89.92 ± 0.78	89.90 ± 0.62
$1NN_{DTW}^{IPL}$	92.01 ± 0.78	92.65 ± 0.78	93.14 ± 0.89	93.76 ± 0.69
$GBSSL_{DTW}^{IPL}$	90.39 ± 0.87	91.08 ± 0.84	92.04 ± 0.51	93.15 ± 0.90
TapNet	76.88 ± 1.89	82.49 ± 1.64	84.83 ± 1.63	86.89 ± 1.29
$ResNet$	<u>94.08</u> ± 0.46	<u>96.28</u> ± 0.85	<u>97.52</u> ± 0.26	<u>97.33</u> ± 1.18
$ResNet^{IPL}$	**97.25** ± 0.61	**97.40** ± 0.60	**98.03** ± 0.49	**98.23** ± 0.61

Table 6: F-Measure results over the SpeechCommand benchmark varying the amount of labelled examples per class. We report average and standard deviation. Bold and underlined text indicate best and second-best results, respectively.

	10	15	20	25
$1NN_{DTW}$	8.27 ± 0.79	8.67 ± 0.83	9.02 ± 0.70	9.54 ± 0.85
$GBSSL_{DTW}$	11.82 ± 0.38	12.81 ± 0.35	13.54 ± 0.32	14.00 ± 0.29
$1NN_{DTW}^{IPL}$	28.85 ± 0.93	29.81 ± 1.25	29.73 ± 0.54	30.30 ± 0.57
$GBSSL_{DTW}^{IPL}$	14.42 ± 0.54	14.95 ± 0.30	15.39 ± 0.36	15.62 ± 0.31
TapNet	12.86 ± 0.62	13.52 ± 0.73	13.70 ± 0.46	14.28 ± 0.40
$ResNet$	<u>46.29</u> ± 3.66	<u>58.51</u> ± 3.05	<u>67.96</u> ± 1.42	<u>73.59</u> ± 0.57
$ResNet^{IPL}$	**59.29** ± 4.42	**69.36** ± 1.46	**74.91** ± 1.52	**78.37** ± 0.67

on three benchmarks ($GTZAN$, $JapWovel$ and *SpeechCom*) while on the rest of the datasets (*Dordogne* and HAR) the behaviours are comparable. In addition, we can observe that $ResNet^{IPL}$ systematically outperforms the recent TapNet approach over all the considered benchmark no matter the amount of labelled samples we consider as initial training set.

Interestingly, we can note that $ResNet$ and $ResNet^{IPL}$ always take advantage when the quantity of labeled samples increases compared to all the other competing approaches. This phenomena is clearly evident considering the $SpeechCom$ benchmark (Table 6). Here, we can see that $ResNet$ and $ResNet^{IPL}$ generally ameliorate their behavior when more labeled samples are available.

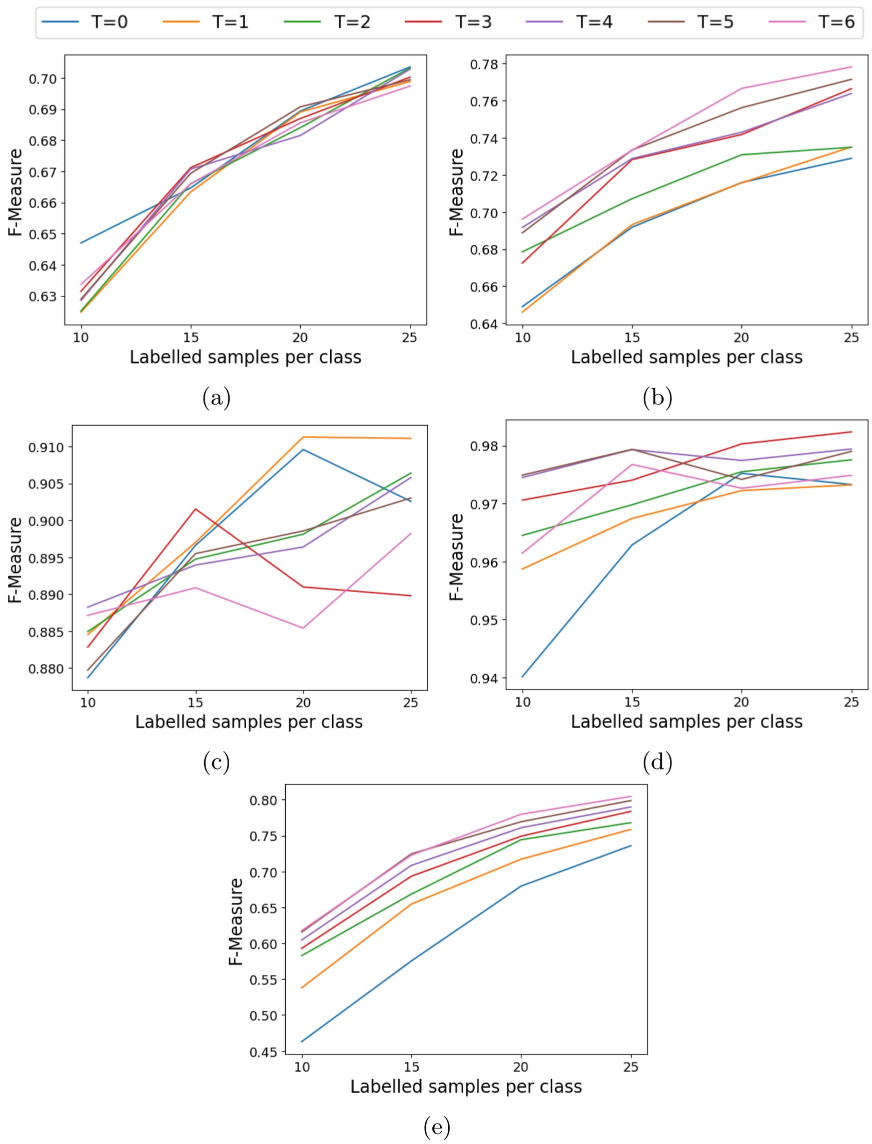


Fig. 1: Sensitivity analysis of $ResNet^{IPL}$w.r.t. the T parameter, varying the amount of available training data, on the considered benchmarks: (a) Dordogne (b) GTZAN (c) HAR (d) JapVowel and (e) SpeechCom.

Figure 1 reports the sensitivity analysis of $ResNet^{IPL}$ regarding the T parameter (the number of iterations of the incremental pseudo labeling process). More in detail, we vary the T parameter in the interval $[0, 6]$, where $T = 0$ corresponds to the behaviour of $ResNet$. Consistently with the results reported in Tables 2, 3, 4, 5, 6, we can observe two different types of behaviors. Regarding $GTZAN$, $JapWovel$ and $SpeechCom$ (resp. Figure 1b, Figure 1d and Figure 1e), we can note that increasing the number of iterations (T) results, in general, in higher value of F-Measure. This is particularly evident for small training data (with a number of labeled samples per class equal or lesser than 15). A different behavior is exhibited by the *Dordogne* and HAR datasets where the parameter T of the incremental pseudo labeling procedure does not really influence the obtained performances in terms of F-Measure.

To sum up the obtained findings, we highlight that, also when extreme low-data regime scenarios are considered, deep learning approaches still exhibit high performances for the classification of multi-variate varying-length time series when compared to standard methods; and, the incremental pseudo labeling strategy clearly ameliorates the results, in terms of F-Measure, considering three benchmarks over five, while on the remaining test cases the performance are comparable w.r.t. the ablation variant that does not involve pseudo labeling.

3.4 Visual Inspection

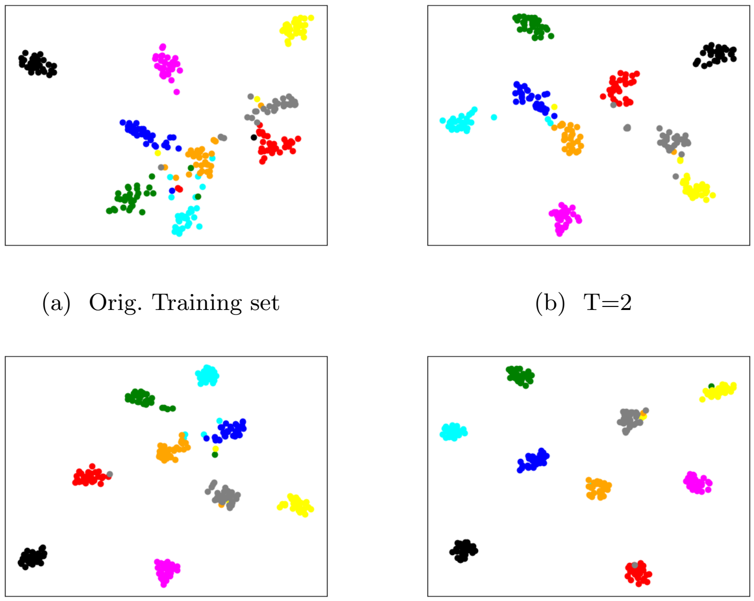
Figure 2 depicts the visualization of the embeddings obtained considering the model trained on the original training set (Figure 2a) and, successively, by $ResNet^{IPL}$ for the values 2, 4 and 6 (Figure 2b, Figure 2c, and Figure 2d, respectively) of the T parameter (the number of iterations in the iterative pseudo labeling process) for the *JapWovel* benchmark.

The original training set involves 10 labeled samples per class. The embeddings are obtained considering the output of the last convolutional layer. We visualize 30 samples per class coming from the test data by means of the two dimensional projection supplied by the T-SNE method [8]. We underline that the same set of test samples is considered over the four different cases. Each colour represents a different class.

We can clearly observe that as the T parameter increases, the cluster structure associated to the underlying data distribution emerges. While the visualisation related to the embeddings obtained by the model trained on the original training set (Figure 2a) exhibits evident confusion among most of the classes, we can note that, the incremental pseudo labeling procedure allows to reduce confusions and to recover a clear cluster structure. More in detail, we can see that, when a value of $T = 4$ (Figure 2c) is considered, the majority of confusions disappear.

4 Conclusion

In this paper, we proposed a framework that combines CNN and self-training to deal with multi-variate time series classification considering low-data regime



(a) Orig. Training set (b) T=2

(c) T=4 (d) T=6

Fig. 2: T-SNE Feature visualization of the same 30 per-class test samples belonging to the *JapWovel* dataset considering the representation learnt from (a) the original (10 labelled samples per class) training data (b) the proposed framework with T=2 (c) the proposed framework with T=4 and (d) the proposed framework with T=6.

scenarios. The proposed framework works in a transductive fashion and it leverages the entropy associated to the classifier prediction to select new samples to enlarge the training set. The evaluation on real-world benchmarks has demonstrated the effectiveness of $ResNet^{IPL}$ w.r.t. recent classification framework and, more generally, the value of deep learning-based strategies to deal with low-data regime scenarios in the context of multi-variate time series classification. Possible follow-ups of our work can be related to the evaluation of recent Transformer [15] models to replace the ResNet internal classifier as well as the test and deployment of the proposed framework in an inductive setting where the goal is the classification of new unseen time series data.

References

1. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J.,

Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
2. Castellanos, M.G., 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**(2), 493–528 (2018)
3. Chen, Y., Hu, B., Keogh, E.J., Batista, G.E.A.P.A.: DTW-D: time series semi-supervised learning from a single example. In: KDD. pp. 383–391. ACM (2013)
4. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.: Deep learning for time series classification: a review. Data Min. Knowl. Discov. **33**(4), 917–963 (2019)
5. Frank, J., Mannor, S., Pineau, J., Precup, D.: Time series analysis using geometric template matching. IEEE Trans. Pattern Anal. Mach. Intell. **35**(3), 740–754 (2013)
6. Gbodjo, Y.J.E., Ienco, D., Leroux, L.: Toward spatio-spectral analysis of sentinel-2 time series data for land cover mapping. IEEE Geosci. Remote. Sens. Lett. **17**(2), 307–311 (2020)
7. Geler, Z., Kurbalija, V., Radovanovic, M., Ivanovic, M.: Comparison of different weighting schemes for the knn classifier on time-series data. Knowl. Inf. Syst. **48**(2), 331–378 (2016)
8. van der Maaten, L., Hinton, G.: Visualizing Data Using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
9. Marussy, K., Buza, K.: SUCCESS: A new approach for semi-supervised classification of time-series. In: ICAISC. vol. 7894, pp. 437–447. Springer (2013)
10. Saporta, A., Vu, T., Cord, M., Pérez, P.: ESL: entropy-guided self-supervised learning for domain adaptation in semantic segmentation. CoRR **abs/2006.08658** (2020)
11. de Sousa, C.A.R., Rezende, S.O., Batista, G.E.A.P.A.: Influence of graph construction on semi-supervised learning. In: ECML/PKDD. vol. 8190, pp. 160–175 (2013)
12. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2005)
13. Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., Yurchak, R., Rußwurm, M., Kolar, K., Woods, E.: Tslearn, a machine learning toolkit for time series data. Journal of Machine Learning Research **21**(118), 1–6 (2020)
14. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. **42**(2), 245–284 (2015)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS. pp. 5998–6008 (2017)
16. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.J.: Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Discov. **26**(2), 275–309 (2013)
17. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: IJCNN. pp. 1578–1585. IEEE (2017)
18. Yamaguchi, Y., Faloutsos, C., Kitagawa, H.: Camlp: Confidence-aware modulated label propagation. In: SDM. pp. 513–521. SIAM (2016)
19. Zhang, X., Gao, Y., Lin, J., Lu, C.: Tapnet: Multivariate time series classification with attentional prototypical network. In: AAAI. pp. 6845–6852 (2020)