A Deep Neural Network Framework for Multivariate Time Series Classification With Positive and Unlabeled Data

Dino Ienco

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2023, accepted 27 February 2023, date of publication 1 March 2023, date of current version 6 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3251194

RESEARCH ARTICLE

A Deep Neural Network Framework for Multivariate Time Series Classification With Positive and Unlabeled Data

DINO IENCO, (Member, IEEE)
INRAE, UMR TETIS, University of Montpellier, 34090 Montpellier, France
e-mail: dino.ienco@inrae.fr

ABSTRACT Positive and unlabelled (PU) learning for multi-variate time series classification refers to build a binary classification model when only a small set of positive and a large set of unlabelled samples are accessible at training stage. Different from binary semi-supervised scenario in which the training set contains labelled samples from both positive and negative classes, in the PU learning setting, only positive samples are labelled due to cost-restriction or issues related to defining what belongs to the negative class. With the objective to deal with this challenging task, here, we propose a new deep learning framework, referred as *DMTS-PUL*. Our method has two different steps: firstly, it selects a set of reliable negative samples from the set of unlabelled data and, successively, it iteratively enriches the training data by selecting pseudo-labels to train a binary classification model via self-training. Experimental evaluations on several benchmarks have highlighted the quality of *DMTS-PUL* w.r.t. competing approaches and the obtained findings have pointed out the suitability of our proposal when only small amounts of positive labelled samples are available.

INDEX TERMS Positive unlabeled learning, multi-variate time series, self-training, recurrent neural network.

I. INTRODUCTION

Nowadays, huge amount of data is being produced by a large and diverse family of sensors (e.g., remote sensors, biochemical sensors, wearable devices and IoT). These sensors typically gather multiple variables over time, resulting in an information flow that can be profitably structured as multivariate time series.

Standard supervised classification methods for multivariate time series classification make the assumption that the training data is fully annotated thus requiring an apriori labelling process which is both costly and time-consuming. Nevertheless, in practical scenarios the speed at which time series are collected often makes unfeasible and unrealistic obtaining labels for both positive and negative classes [6].

On the other hand, when experts are required to provide positive and negative labels for a binary classification task, it can happen that the concept of positive sample is clearly defined while the idea of negative sample is not well-established [38]. As a consequence, only a small portion of a so-constituted training set is labelled.

In such scenarios, the available training data is constituted by a set of positive samples together with a set of unlabelled ones spanning both positive and negative concepts. Such a setting is known as learning from positive and unlabelled (PU) data [2]. More precisely, this setting differs from standard supervised or semi-supervised classification by the absence of labelled negative samples in the training set.

The concept of learning from PU data fits within the increasing interest in developing weakly supervised learning frameworks [39], such as learning from positive-only or one-class data [16] and semi-supervised learning [14], that relax the strict requirement related to the access of fully annotated datasets. Learning from PU data differs from one-class classification since it explicitly involves unlabelled data in the learning process while, it specialises semi-supervised learning dealing with the case in which label information for all the classes is not available. In literature, different

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

approaches were already proposed to deal with PU learning for tabular/propositional data defining the problem as a cost-sensitive task [7], [21], conceiving two-step strategies where reliable negative samples are first selected and then used to train a traditional binary classifier [10] with self-training [1] or model the unlabeled data as negative samples with label noise [26].

Focusing on time series data, PU learning can model real world problems coming from application domains like multimedia, medicine, aerospace, finance, manufacturing, entertainment and remote sensing [27]. Reference [27] introduces a learning approach to deal with time series information under the lens of learning from PU data setting. The proposed method is based on a clustering-based solution to identify reliable negative samples. The clustering procedure is coupled with euclidean distance. Finally, once positive and reliable negative samples are available, the classification is performed by means of one nearest neighbours classifier (1NN). Another framework based on 1NN classifier is presented in [6]. Here, the authors propose a framework in which the positive set is incrementally expanded by means of a similarity measure combining both euclidean distance and dynamic time warping [25]. Recently, [5] and [22] still highlight the 1NN classifier, coupled with dynamic time warping, as main solution to deal with time series classification under the learning from PU data setting. Unfortunately, all such approaches are mainly devoted to deal with univariate time series while there is a lack of dedicated research studies to cope with the complexity of multi-variate time series even if they are becoming predominant in everyday scenarios. Despite several research works [3], [13], [33] have highlighted the suitability of deep learning based techniques for multi-variate time series, to the best of our literature survey, no research study is yet conducted to address the multi-variate time series classification task when only positive and unlabelled data is available through deep learning approaches.

To tackle this point, here, we propose a new strategy, named *DMTS-PUL* (Deep neural network-based Multi-variate Time Series framework for Positive and Unlabelled Learning) to cope with multi-variate time series classification when only positive and unlabelled data is available. *DMTS-PUL* is a two stage framework where, firstly, a deep autoencoder, based on Recurrent Neural Networks (RNN), selects and identifies reliable negative time series samples and, secondly, an iterative pseudo labelling procedure is conceived to learn a binary classification model via self-training. The contributions of this work are: i) a new framework, named *DMTS-PUL* to deal with multi-variate time series classification in presence of only positive and unlabelled data and; ii) a strategy to incrementally exploit unlabelled samples to enrich the training set (both positive and reliable negative data) via a self-training and pseudo-labelling procedure.

The rest of the paper is structured as follows: the *DMTS-PUL* framework is introduced in Section II, experimental settings as well as experimental evaluations are detailed in Section III while Section IV concludes and draws future works.

II. METHODOLOGY

In this section, we introduce *DMTS-PUL* (Deep neural network-based Multi-variate Time Series framework for Positive and Unlabelled Learning), a framework to deal with multi-variate time series classification from only positive and unlabelled data. Figure 1 provides an overview of *DMTS-PUL*.

Our framework is composed by two steps. A first step (Top of Figure 1) dedicated to select reliable negative samples that highly differ from those belonging to the positive set. This step is addressed via a Recurrent Neural Network (RNN) autoencoder that is used to model the set of positive samples. In the second step (Bottom of Figure 1), *DMTS-PUL* learns a multivariate time series classifier, based on one dimensional convolutional neural network [9], to deal with the binary classification problem. In this stage, an iterative pseudo labelling (IPL) procedure based on self-training is designed to enrich the labelled data on which the model is trained on.

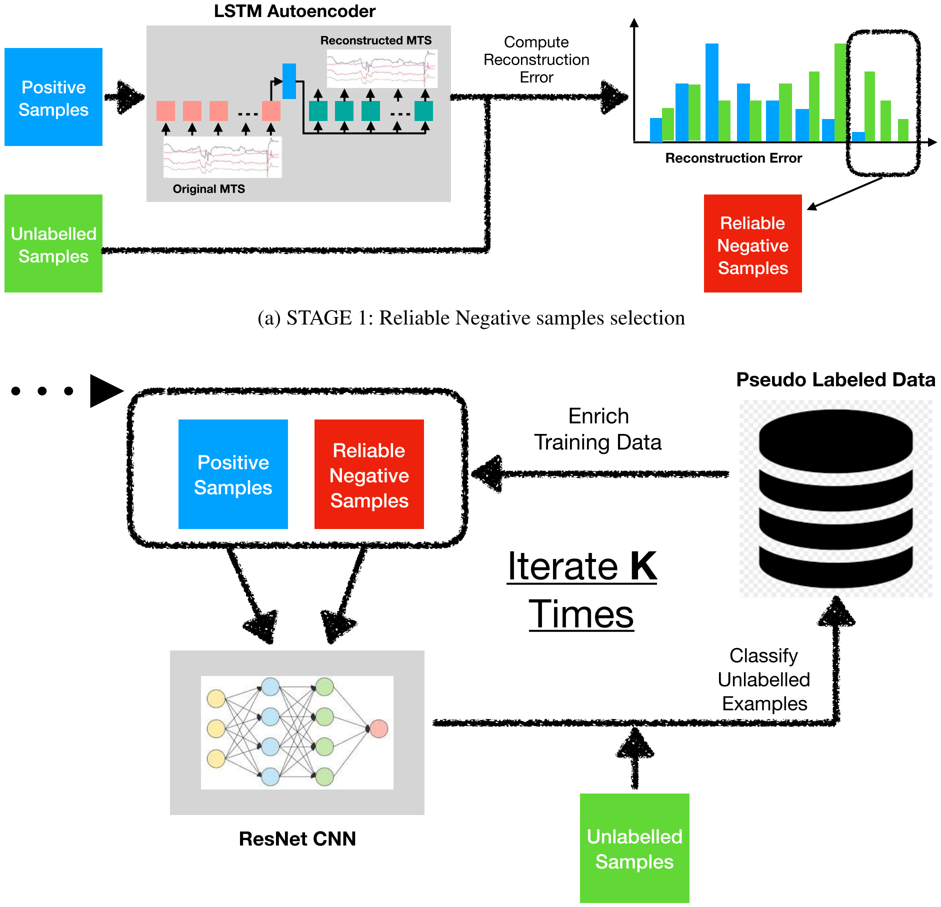
A. RELIABLE NEGATIVE SAMPLES SELECTION

The first step of *DMTS-PUL* copes with the identification of a set of reliable negative (RN) samples. To this end, we first model the positive set of multivariate time series via an RNN autoencoder then, we use the learnt neural network to rank the samples from the unlabelled set and select reliable negative multi-variate time series data. To rank unlabelled samples, the reconstruction error is used as measure. Samples associated with a low reconstruction error probably belong to the positive class while, samples related to high reconstruction error are likely to belong to the negative class. Finally, multivariate time series with the highest reconstruction errors, coming from the unlabelled set, are identified as reliable negative samples.

The use of Recurrent Neural Network naturally fits our setting since this model is especially adapted to manage sequential data [23] and, in particular, compressing sequential information to extract useful multi-variate time series representations [13], [17]. The most well-known type of RNN is the Long-Short Term Memory (LSTM) [12] model that was introduced to learn long term dependencies as well as cope with the vanishing and exploding gradient issues characterizing previous RNN architectures [12].

In our framework, we use the LSTM neural network to encode and, subsequently, decode the set of positive multi-variate time series samples with the aim to build an (autoencoder) model especially tailored to compress and reconstruct multi-variate time series coming from the positive set.

Figure 2 sketches the structure of the LSTM autoencoder. For an element at timestamps t returned by the LSTM autoencoder, we perform an extra linear transformation to generate the reconstructed element $\hat{x}_t$, since, for regression tasks, the softmax layer is replaced with a fully connected layer without



(a) STAGE 1: Reliable Negative samples selection

(b) STAGE 2: Iterative Pseudo Labelling procedure

FIGURE 1. The *DMTS-PUL* framework. It has two stages: (a) the selection of reliable negative multi-variate time series samples and (b) the iterative pseudo labelling procedure. The former stage allows to extract a set of reliable negative samples that are intrinsically different from the samples belonging to the positive class. The latter stage incrementally trains a classifier from the sets of positive and reliable negative multi-variate time series data via self-training to deal with the underlying binary classification problem.

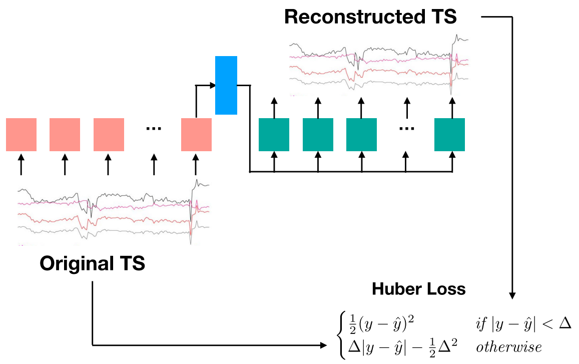

Reconstructed TS
Original TS
Huber Loss
$\begin{cases}\frac{1}{2}(y-\hat{y})^2 & if\ |y-\hat{y}|<\Delta \\ \Delta|y-\hat{y}|-\frac{1}{2}\Delta^2 & otherwise\end{cases}$

FIGURE 2. The LSTM autoencoder structure. The model takes as input the original time series (TS) and tries to reconstruct the same signal (reconstructed TS). The reconstruction error is computed via the Huber Loss.

any activation function [20]. Finally, the autoencoder model is trained in a end-to-end manner through a reconstruction loss computed only considering the set of positive labelled multivariate time series:

$$LOSS = \frac{\sum_{TS \in P} HLoss(TS, Linear(LSTM-AE(TS)))}{|P|} \tag{1}$$

where $(Linear(LSTM-AE(TS)))$ is the LSTM autoencoder followed by linear transformation and HLoss is the Huber Loss [20].

defined as follows:

$$HLoss(y,\hat{y}) = \begin{cases}\frac{1}{2}(y-\hat{y})^2 & if\ |y-\hat{y}| < \Delta \\ \Delta|y-\hat{y}| - \frac{1}{2}\Delta^2 & otherwise\end{cases} \tag{2}$$

The Huber loss exhibits the same behaviour of the mean squared error (MSE) loss when the error is lower than a specified threshold Δ while it behaves as the mean absolute error (MAE) loss otherwise. Additionally, it is differentiable at 0. The Huber loss allows to avoid standard limitations related to MAE and MSE functions as well as retains good properties of the two standard regression losses. More in detail, differently to MAE, it circumvents large gradient backpropagation when the estimated quantity is getting closer to the real value and, conversely from the MSE loss, it is more robust to possible outliers. In our scenario we set Δ equals to 1 for the Huber Loss.

Once the LSTM autoencoder is trained on the set of positive multivariate time series, then, it is used on the set U of unlabelled time series. For each multi-variate time series $TS \in U$, a reconstruction error is computed and the samples with the highest reconstruction errors are selected to build the Reliable Negative set (RN). The reconstruction error is computed by means of the Huber Loss in order to be coherent with the way in which the LSTM autoencoder was learnt. Finally, we impose that the RN set size is equal to the size of the positive sample set P.

B. ITERATIVE PSEUDO LABELLING PROCEDURE

Once both P and RN sets are available, we train a classification model to deal with the underlying binary classification task. More precisely, the classification model is learnt via an iterative pseudo labelling (IPL) procedure, reported in Algorithm 1. The objective is to iteratively enrich the set of positive P and reliable negative RN samples exploiting the set of unlabelled multi-variate time series data U, via self-training.

The procedure depicted in Algorithm 1 takes as input the set of positive samples (P), the set of reliable negative samples (RN), the set of unlabelled multi-variate time series (U), the number of iterations (K) and the amount of samples to be added to the positive (resp. reliable negative) set at each iteration (l). The output of the algorithm is a binary multi-variate time series classification model that is trained over the enriched set of positive and reliable negative samples.

At the beginning, the classification model ($Classifier$) is initialized and then trained on the set of positive P and reliable negative RN samples (Line 1-2). Then, the incremental process starts (Line 4-13). At each iteration, the classification model is applied on the set of unlabelled multi-variate time series U and, for each $TS \in U$, a binary class distribution is obtained (Line 5). Subsequently, the class distribution jointly with the unlabelled set U, the amount of new samples to enrich the current training dataset l and the information about the current iteration i are taken as inputs from the $SampleSel$ procedure. This procedure (Line 6) selects a bunch of new positive multi-variate time series ($newP$) and the new set of reliable negative samples (RN). Then, the positive set of training data as well as the set of unlabelled samples is updated (Line 7-8) and a new binary classification model is trained over the current set of positive and reliable negative multi-variate time series samples from scratch (Line 9-10). We underline that, at each iteration, the set of reliable negative sample RN is completely updated (Line 6). This is done with the objective to recover possible mistakes done in the early rounds of the iterative pseudo labelling procedure thus, reducing possible confirmation bias [31].

Algorithm 1 Incremental Pseudo Labeling Procedure

```
Require: $P$ (set of positive samples), $RN$ (set of reliable negative samples), $U$ (unlabelled multi-variate time series), $K$ (number of iterations), $l$ (samples to be added at each iteration).
Ensure: $Classifier$.
 1: $Classifier \leftarrow$ initModel()
 2: $Classifier \leftarrow$ TrainModel( $Classifier$, $P$, $RN$)
 3: $i \leftarrow 0$
 4: while i < K do
 5:     classDistrib $\leftarrow$ Classify($Classifier$, $U$)
 6:     $newP$, $RN$ $\leftarrow$ SampleSel($U$, classDistrib, l, i)
 7:     $P \leftarrow P \cup newP$
 8:     $U \leftarrow U \setminus newP$
 9:     $Classifier \leftarrow$ initModel()
10:     $Classifier \leftarrow$ TrainModel($Classifier$, $P$, $RN$)
11:     $i \leftarrow i + 1$
12: end while
13: return $Classifier$
```

Concerning Algorithm 1, two points must be defined: firstly, the classification model and, secondly, how the sample selection procedure is implemented.

For the classification model, we base our choice on the recent findings reported by [9]. Among several deep learning architectures for multi-variate time series classification, the Convolutional Residual Network *ResNet* model proposed in [36] has been pointed out as the best current model to cope with the complexity of time series information through one dimensional convolutions. Due to this fact, we choose such model as classification backbone in our work.

The second point involves the definition of a sample selection strategy. Such a strategy, ($SampleSel(\cdot)$), is mainly based on the analysis of the class distribution outputted by the classification model. More in detail, for each sample TS we exploit the class distribution $pd(TS)$ outputted by the classification model. $pd(TS)$ is the probability distribution over the two possible classes that corresponds to the softmax output of the classification model regarding the sample TS. Our strategy selects unlabelled samples on which the classifier has the highest confidence. To this purpose, we consider as surrogate of the confidence measure the entropy on the classifier output. The entropy measure is defined as:

$$H(TS) = - \sum_{c \in (P,RN)} pd_c(TS) \times \log(pd_c(TS)) \qquad (3)$$

This measure has already demonstrated its quality in pseudo labelling strategies to select valuable samples in the context of image analysis and semantic segmentation [29].

Samples with low entropy values correspond to time series on which the classifier has high confidence in its prediction. At each iteration of the Algorithm 1, the entropy measure is employed to select l new positive samples ($newP$) and the $l \times i$ reliable negative samples. The former will be added to the current set of positive labelled samples while the latter will replace the current set of reliable negative (RN) samples on which the classification model will be trained.

III. EXPERIMENTS

In this section we present and discuss the experimental evaluation we have designed to assess the performance of *DMTS-PUL*. To this end, we compare *DMTS-PUL* with different competing methods over several benchmarks and we quantitatively and qualitative inspect the behaviour of our proposal.

A. COMPETITORS

As competitor, we consider the following methods:

- The widely-adopted PU learning approach for time series data based on the 1NN classifier [6]. We consider two variants: one based on Dynamic Time Warping [25], namely $1NN-DTW$, and another one based on Euclidean distance, referred as $1NN-EUCL$.
- The probabilistic PU method proposed in [8]. We couple the probabilistic framework with the Random Forest classifier [4] with a number of internal trees equals to 300. We name this approach $RF-PUL$.
- The non-negative risk estimator for positive unlabelled learning introduced in [19]. We couple the non-negative risk estimator with the Convolutional Residual Network *ResNet* model proposed in [36]. We refer to this method as $nnPU$.
- A recent framework, proposed in [15], for positive and unlabelled learning classification that leverages non-negative risk estimator and it explicitly captures the existence of possible selection bias in the labelling process. We refer to this approach as *nnPUSB*.
- The One-Class Support Vector Machine methods. Learning from PU data can be addressed considering only positive labelled samples. We name this approach $OCSVM$.
- An ablation of our framework, where the incremental pseudo labelling procedure is discarded. We name this approach $DMTS\text{-}PUL^{NoIPL}$.

B. DATA AND EXPERIMENTAL SETTINGS

The evaluation has been carried out on seven benchmarks [9] coming from disparate application domains and characterized by contrasted features in terms of number of samples, number of attributes (dimensions) and time series lengths: *ArabDigits*,[1] *Dordogne*, *ECG5000* [1], *HAR*,[2] *PenDigits* [1], *seqMNIST*[3] and *SpeechCom*.[4] All datasets, except *Dordogne* – which was obtained contacting the authors of [11], are available online. To have a fair comparison among the different methods and with the aim to provide a controlled scenario for the positive unlabelled task evaluation, for each benchmark, we only consider the two majority classes (the two classes with the highest number of samples) to set up an underlying binary classification task. The majority class is considered as positive while the second more represented class is considered as negative class. Table 1 reports the benchmark characteristics after the selection of the two majority classes.

TABLE 1. Benchmarks Characteristics.

Dataset	# Samples	# Dims	Min/Max Length	Avg. Length
ArabDigits	1 760	13	9/87	40
Dordogne	4 257	6	23/23	23
ECG5000	4 686	1	140/140	140
HAR	3 850	9	128/128	128
PenDigits	2 288	2	8/8	8
seqMNIST	15 170	28	28/28	28
SpeechCom	4 757	40	14/32	31

For each dataset, we split the sample set in two partitions: training and test. Each partition involves 50% of the original benchmark. After that, the training partition is divided again in two: the positive and the unlabelled partitions. While the former is composed of only positive samples, the latter contains samples coming from both the positive and the negative classes. Due to the fact that the quantity of positive samples can influence the behaviour of the machine learning model, we consider increasing amount of positive samples ranging in the set $\{30, 60, 90, 120, 150\}$. This means that, when 120 samples are considered as positive, the rest of the train data is treated as the unlabeled set. Varying the amount of positive samples also permits to assess the stability of the approaches w.r.t. the amount of (positive) knowledge it can access. Once a model is trained, the independent test set is used to evaluate the prediction performances.

As evaluation metric we choose the F-Measure [30]. The F-Measure is defined as the harmonic mean between the Precision and Recall measures and it supplies a general information summarizing both true and false positive rates. Due to the non deterministic nature of the sample selection process, the obtained results are averaged over five different trials for each method and benchmark.

DMTS-PUL is based on two different stages: (a) the selection of reliable negative multi-variate time series samples and (b) the iterative pseudo labelling procedure. For the former stage, the LSTM autoencoder model is trained for 300 epochs with a batch size equals to 8 and a learning rate of 5×10^{-3} through the RMSprop optimizer.[5] The LSTM autoencoder has a number of unit equals to 128. In addition, the autoencoder model is trained via a denoising strategy [24] to boost its robustness (noise is injected via a Gaussian multiplicative noise with mean equals to 1 and standard deviation equals to 0.05). For the second stage, we exploit the *ResNet*-based classification described in [9]. To this end, we use its public available implementation.[6]

[1] http://www.timeseriesclassification.com/index.php

[2] https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[3] https://paperswithcode.com/sota/sequential-image-classification-on-sequential

[4] https://www.tensorflow.org/datasets/catalog/speech_commands

[5] http://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf

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

The multi-variate time series ResNet classifier is trained for 300 epochs via the Adam optimizer [18] with a learning rate of × 10^{-3} and a batch size equals to 4. The same settings are employed to training the ResNet classifiers associated to the *nnPU* and *nnPUSB* competing methods. Furthermore, since we exploit an iterative pseudo labelling procedure, we have to set the number of iteration (K) as well as the number of samples to integrate at each step (l). In the main experiments, we set the number of iterations equals to 3 and the number of samples to integrate at each round (for both the positive and the reliable negative classes) equal to 30.

Regarding the competing methods, except for $1NN - DTW$ that naturally manages time series with varying length, we perform zero padding for the *ArabDigits* and the *SpeechCom* benchmarks. Concerning the methods based on the one nearest neighbour classifier, following the procedure proposed in [5], [6], and [22] where positive examples are integrated as the process going on, we integrate the same amount of positive samples as the one that is integrated in the incremental pseudo labelling procedure associated to our approach. This result in 90 new samples that are integrated by the 1NN based methods. *DMTS-PUL* is implemented via the Tensorflow 2 python library while the implementation of competing methods is based on TSLEARN [32] and SCIKIT-learn python libraries [28].

C. QUANTITATIVE RESULTS

Figure 3 reports the results obtained on the different benchmarks by the competing methods varying the amount of positive labelled data. We can observe that *DMTS-PUL* always outperforms all the competing approaches over all the seven involved datasets.

Considering the competitors, they almost share a similar behaviour over all the datasets. The least effective approaches are the ones based on the one nearest neighboors classifier and the one based on the non-negative risk estimation principle with selection bias, *nnPUSB*. Regarding the former group, the ones based on the 1NN classifier, despite the high level performances of such approach for univariate time series classification task [5], [6], [22], they have serious issues when multi-variate time series are considered. The *nnPU* method, differently from all the other approaches, exhibits a varying behaviour depending to the dataset. While it shows interesting performances on *ArabDigits* and *SpeechCom*, it achieves poor results over all the other benchamrks. The *OCSVM* method, that only leverages positive labelled samples, generally exhibits a good average performance in terms of F-Measure. Finally, on six over seven cases, the best competing approach is represented by the Random Forest classification method coupled with the positive and unlabelled learning framework proposed in [8]. This approach also shown competitive performances on the *ArabDigits* dataset while it is largely far from the performances obtained by *DMTS-PUL* on all the remaining multi-variate time series datasets. Regarding the direct comparison between *DMTS-PUL* and its ablation (*DMTS-PUL*^{NoIPL}), we can note that *DMTS-PUL* achieves general better performances than its ablation for small amount of positive labelled data (between 30 and 90). This is particularly evident on the *ArabDigits*, *Dordogne* and *SpeechCommand* benchmarks. When the size of the available positive multi-variate time series set is above 90, the performances of the two approaches are comparable but *DMTS-PUL* behaves slightly better than its ablation on the majority of the benchmarks. In addition, we can also observe that *DMTS-PUL* achieves quite stable performances no matter the amount of positive labelled data it can access. This is not the case for the majority of the competing approaches.

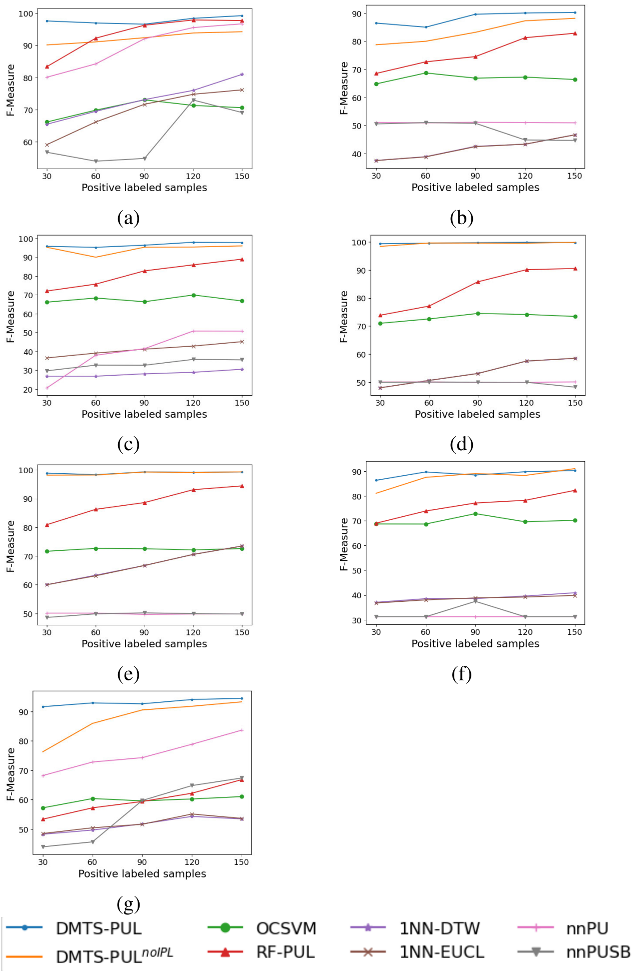


FIGURE 3. Average F-Measure of the different approaches, varying the amount of labelled data, on: (a) ArabDigits (b) Dordogne (c) ECG5000 (d) HAR (e) PenDigits (f) seqMNIST and (g) SpeechCom.

The comparison between *DMTS-PUL* and *DMTS-PUL*^{NoIPL} highlights the quality of the IPL procedure involved in our framework. The iterative pseudo labelling procedure makes *DMTS-PUL* well suited for low-data regime when only a small amount of labelled data is available. Moreover, the obtained findings also suggested that our framework does not require enormous volume of data to obtain competitive results thus positively impacting the time-consuming and human-effort labelling task prior to any classification scenario.

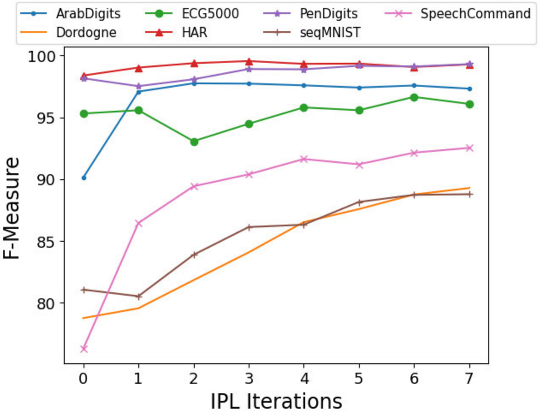


FIGURE 4. Sensitivity analysis of *DMTS-PUL* w.r.t. the *K* parameter with 30 positive labelled samples.

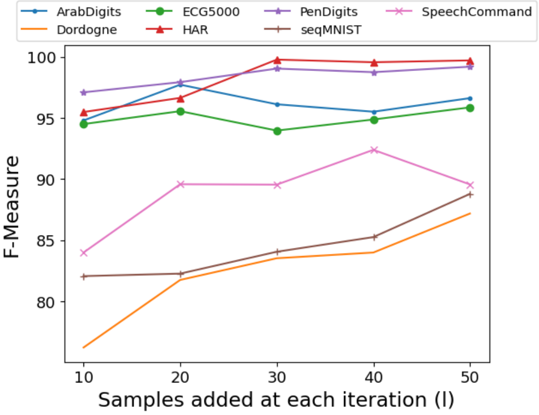
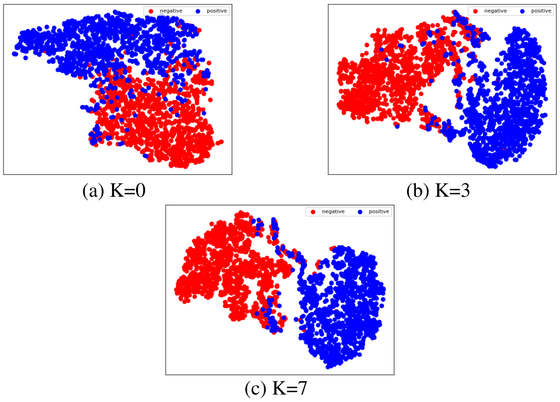


FIGURE 5. Sensitivity analysis of *DMTS-PUL* w.r.t. the *l* parameter with 30 positive labelled samples and a total of 3 iterations ($K = 3$).

Figure 4 depicts a sensibility analysis of our framework varying the number of iterations in the IPL procedure. We vary such value from 0 (no IPL procedure is performed, these results are equivalent to the ones obtained by the *DMTS-PUL*^{NoIPL}) to 7. For this experiment, a set of positive samples with a size equal to 30 is considered. Generally, as we can expect, we observe that as the number of iterations increases, the F-Measure performances ameliorate. The involved benchmarks are characterized by two distinct behaviours. A first one shared by *ArabDigits*, *ECG5000*, *HAR* and *PenDigits*. Here, the performances stack and remain stables after three or four iterations of the IPL procedure. Conversely, for *Dordgone*, *seqMNIST* and *SpeechCommand* we see that the performances are still growing as the number of iterations increases. This is probably due to the fact that, while on the former bunch of datasets *DMTS-PUL* already obtains high level performances (more than 95 points of F-Measure), the second set of test cases exhibit higher complexity/difficulty thus, resulting in general lower absolute performances of *DMTS-PUL* with room for improvement.

Figure 5 depicts a sensibility analysis of our framework varying the amount of samples to be added to the positive (resp. reliable negative) set at each iteration. We vary such a value from 10 to 50. For this experiment, we set the parameter K equal to 3. We can observe two different kind of behaviour. The first one, shared by *PenDigits*, *HAR*, *ECG5000*, *Arabdigits* and *SpeechCommand*, where stable performances are achieved starting from a value of the l parameter equals or greater than 30. Conversely, for the remaining benchmarks, *Dordogne* and *seqMNIST*, our framework achieves better performances as the value of the parameter l increases.



(a) K=0

(b) K=3

(c) K=7

FIGURE 6. T-SNE features visualization of the test samples belonging to *SpeechCom* considering the representation learnt from (a) the original training data (*P* and *RN*) obtained after the first stage of our framework (*K*=0) (b) after three rounds (*K*=3) and (c) after seven rounds (*K*=7) of the IPL procedure. The initial positive labelled set contains 30 samples.

D. VISUAL INSPECTION

Figure 6 depicts the visualization of the embeddings obtained by the binary classification model on the *SpeechCom* benchmark varying the number of iterations (K) of the iterative pseudo labelling procedure: no IPL procedure (Figure 6(a)), $K = 3$ (Figure 6(b)), and $K = 7$ (Figure 6(c)). Also in this case, the initial set P involves 30 multi-variate time series samples.

The embeddings are obtained considering the output of the last convolutional layer. We visualize the whole set of test data by means of the two dimensional projection supplied by the T-SNE method [34]. Each colour represents a different class.

We clearly see that as the number of iterations K increases, the cluster structure associated to the underlying data distribution emerges. While the embedding visualisation related to the classification model learnt on the P and RN sets directly coming from the first stage of *DMTS-PUL* (Figure 6(a)) exhibits visual confusion among the positive and negative classes, we can observe that the IPL procedure allows to reduce confusions and to recover a more clear cluster structure. More precisely, we can note that, when a value of $K = 3$ (Figure 6(b)) is considered, the cluster structure is more visible. Such visual and qualitative inspection confirms the quantitative findings we have reported in the previous section.

IV. CONCLUSION

In this paper, we have proposed *DMTS-PUL*, a framework to deal with multi-variate time series classification tasks when only positive and unlabelled data is available. Our framework, named *DMTS-PUL*, involves two different stages: (a) the selection of reliable negative multi-variate time series and (b) an iterative pseudo labelling procedure to build a binary classification model based on self-training. The evaluation on seven real-world benchmarks has demonstrated the effectiveness of *DMTS-PUL* especially when only a limited amount of (positive) labelled data is considered. Possible future works can be related to: i) assess the proposed framework in more challenging classification tasks where positive and negative concepts are loosely separated from each other, ii) evaluate recent Transformer [35] models as replacement for the ResNet backbone and iii) adapt the proposed framework to cope with a multi-positive unlabelled learning setting [37] where the binary setting is extended to a multi-class scenario.

REFERENCES

[1] T. M. A. Basile, N. D. Mauro, F. Esposito, S. Ferilli, and A. Vergari, ‘‘Density estimators for positive-unlabeled learning,’’ in *Proc. Int. Workshop New Frontiers Mining Complex Patterns*, vol. 10785. Skopje, Macedonia, Sep. 2017, pp. 49–64.

[2] J. Bekker and J. Davis, ‘‘Learning from positive and unlabeled data: A survey,’’ *Mach. Learn.*, vol. 109, no. 4, pp. 719–760, Apr. 2020.

[3] F. M. Bianchi, L. Livi, K. Ø. Mikalsen, M. Kampffmeyer, and R. Jenssen, ‘‘Learning representations of multivariate time series with missing data,’’ *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106973.

[4] L. Breiman, ‘‘Random forests,’’ *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[5] M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, and J. M. Benítez, ‘‘On the stopping criteria for k-nearest neighbor in positive unlabeled time series classification problems,’’ *Inf. Sci.*, vol. 328, pp. 42–59, Jan. 2016.

[6] Y. Chen, B. Hu, E. Keogh, and G. E. A. P. A. Batista, ‘‘DTW-D: Time series semi-supervised learning from a single example,’’ in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 383–391.

[7] C. Elkan, ‘‘The foundations of cost-sensitive learning,’’ in *Proc. IJCAI*, B. Nebel, Ed., 2001, pp. 973–978.

[8] C. Elkan and K. Noto, ‘‘Learning classifiers from only positive and unlabeled data,’’ in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 213–220.

[9] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, ‘‘Deep learning for time series classification: A review,’’ *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.

[10] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu, ‘‘Text classification without labeled negative documents,’’ in *Proc. 21st Int. Conf. Data Eng. (ICDE)*, 2005, pp. 594–605.

[11] Y. J. Eudes Gbodjo, D. Ienco, and L. Leroux, ‘‘Toward spatio–spectral analysis of Sentinel-2 time series data for land cover mapping,’’ *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 307–311, Feb. 2020.

[12] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, ‘‘LSTM: A search space Odyssey,’’ *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[13] D. Ienco and R. Interdonato, ‘‘Deep multivariate time series embedding clustering via attentive-gated autoencoder,’’ in *Proc. PAKDD*, vol. 12084, 2020, pp. 318–329.

[14] D. Ienco and R. G. Pensa, ‘‘Enhancing graph-based semisupervised learning via knowledge-aware data embedding,’’ *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 5014–5020, Nov. 2020.

[15] M. Kato, T. Teshima, and J. Honda, ‘‘Learning from positive and unlabeled data with a selection bias,’’ in *Proc. ICLR*, 2019, pp. 1–17.

[16] S. S. Khan and M. G. Madden, ‘‘One-class classification: Taxonomy of study and review of techniques,’’ *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jan. 2014.

[17] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, ‘‘Outlier detection for time series with recurrent autoencoder ensembles,’’ in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2725–2732.

[18] D. P. Kingma and J. Ba, ‘‘Adam: A method for stochastic optimization,’’ in *Proc. ICLR*, 2015, pp. 1–41.

[19] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, ‘‘Positive-unlabeled learning with non-negative risk estimator,’’ in *Proc. NIPS*, 2017, pp. 1675–1685.

[20] S. Lathuiliere, P. Mesejo, X. Alameda-Pineda, and R. Horaud, ‘‘A comprehensive analysis of deep regression,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2065–2081, Sep. 2020.

[21] W. S. Lee and B. Liu, ‘‘Learning with positive and unlabeled examples using weighted logistic regression,’’ in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, vol. 3, Aug. 2003, pp. 448–455.

[22] S. Liang, Y. Zhang, and J. Ma, ‘‘Active model selection for positive unlabeled time series classification,’’ in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 361–372.

[23] T. Linzen, E. Dupoux, and Y. Goldberg, ‘‘Assessing the ability of LSTMs to learn syntax-sensitive dependencies,’’ *Trans. ACL*, vol. 4, pp. 521–535, Dec. 2016.

[24] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, ‘‘Speech enhancement based on deep denoising autoencoder,’’ in *Proc. Interspeech*, Aug. 2013, pp. 436–440.

[25] A. Mueen and E. Keogh, ‘‘Extracting optimal performance from dynamic time warping,’’ in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 2129–2130.

[26] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, ‘‘Cost-sensitive learning with noisy labels,’’ *J. Mach. Learn. Res.*, vol. 18, pp. 155:1–155:33, Jan. 2017.

[27] M. N. Nguyen, X. Li, and S.-K. Ng, ‘‘Positive unlabeled leaning for time series classification,’’ in *Proc. IJCAI*, 2011, pp. 1421–1426.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ‘‘Scikit-learn: Machine learning in Python,’’ *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

[29] A. Saporta, T.-H. Vu, M. Cord, and P. Pérez, ‘‘ESL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation,’’ 2020, *arXiv:2006.08658*.

[30] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st ed. Boston, MA, USA: Addison-Wesley, 2005.

[31] A. Tarvainen and H. Valpola, ‘‘Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,’’ in *Proc. ICLR*, 2017, pp. 1–16.

[32] R. Tavenard, ‘‘Tslearn, a machine learning toolkit for time series data,’’ *J. Mach. Learn. Res.*, vol. 21, no. 118, pp. 1–6, 2020.

[33] D. J. Trosten, A. S. Strauman, M. Kampffmeyer, and R. Jenssen, ‘‘Recurrent deep divergence-based clustering for simultaneous feature learning and clustering of variable length time series,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3257–3261.

[34] L. van der Maaten and G. Hinton, ‘‘Visualizing data using t-SNE,’’ *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ‘‘Attention is all you need,’’ in *Proc. NIPS*, 2017, pp. 5998–6008.

[36] Z. Wang, W. Yan, and T. Oates, ‘‘Time series classification from scratch with deep neural networks: A strong baseline,’’ in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.

[37] Y. Xu, C. Xu, C. Xu, and D. Tao, ‘‘Multi-positive and unlabeled learning,’’ in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3182–3188.

[38] H. Yu, J. Han, and K. C.-C. Chang, ‘‘PEBL: Positive example based learning for web page classification using SVM,’’ in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2002, pp. 239–248.

[39] Z.-H. Zhou, ‘‘A brief introduction to weakly supervised learning,’’ *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, Aug. 2018.

DINO IENCO (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Torino, Torino, Italy, in 2006 and 2010, respectively. He joined the TETIS Laboratory, IRSTEA, Montpellier, France, in 2011, as a Junior Researcher. His research interests include machine learning, data science, graph databases, social media analysis, information retrieval, and spatio-temporal data analysis with a particular emphasis on remote sensing data and earth observation data fusion. He served in the program committee of many international conferences on data mining, machine learning, and database, including IEEE ICDM, ECML PKDD, ACML, and IJCAI, and served as a reviewer for many international journal in the general field of data science and remote sensing.