SMATE: Semi-Supervised Spatio-Temporal Representation Learning on Multivariate Time Series

Jingwei Zuo, Karine Zeitouni and Yehia Taher
DAVID Lab, University of Versailles, Université Paris-Saclay, Versailles, France
Email: {jingwei.zuo, karine.zeitouni, yehia.taher}@uvsq.fr

Abstract

Learning from Multivariate Time Series (MTS) is an important problem in the Data Mining community, which has attracted wild-spread attention in recent years. In particular, learning from the weak-label MTS is a practical challenge considering the complex dimensional and sequential data structure. The previous work on Semi-supervised Time Series Classification (SMTSC) generally relies on self-training and carefully designed distance, giving the difficulties to control the stopping criterion when introducing the pseudo labels and the unreliability when extended to MTS scenarios where the distance-based classifiers (e.g., DTW_I, DTW_D, DTW_A) may not be strong enough. The emerging deep learning-based approaches are reliable in specific classification tasks but rarely consider the label shortage and explore the characteristics of MTS data thoroughly. In this paper, we propose SMATE, a novel SMTSC model with the Spatio-Temporal representation learned from weakly labeled multivariate time series. Specifically, apart from the temporal dynamic features extracted by a recurrent network, we design a spatial modelling block to capture the spatial dynamic features between each one-dimensional series. The features are embedded into a low-dimensional space via an auto-encoder based structure. A semi-supervised three-step regularization process is proposed on top of the reconstruction objective, to foster the model adaptation of cluster-specific representations. Empirically, we evaluate our method on 22 practical datasets in UEA MTS archive with eleven state-of-the-art baseline methods on fully supervised task, and two baselines on semi-supervised task, the results show the reliability of our proposed method.

Introduction

Most Multivariate Time Series (MTS) data, such as sensor readings, are labeled during the data collection process. The post-labeling on MTS is much more costly than classic data (e.g., image, text, etc.) due to the low interpretability over the real-valued sequence, leading to a considerable constraint for MTS classification in real-life scenarios.

Weakly supervised learning becomes an alternative option of the fully supervised algorithm with the valuable information learned from the unlabeled samples. The previous studies on weak-label Time Series (TS) learning are usually based on self-learning (Wei and Keogh 2006) or Positive Unlabeled Learning (Nguyen, Li, and Ng 2011) (He et al. 2015) with carefully designed distance measure (Chen et al. 2013) or stopping criterion (Ratanamahatana and Wanichsan 2008) to import the pseudo labels. However, they mostly focus on the Univariate Time Series with One-Nearest-Neighbor classifier on raw data space, which is considered as baseline in terms of accuracy (Bagnall et al. 2017) by enormous emerging techniques, such as Deep Neural Networks (DNNs) (Tang et al. 2020) or ensemble methods (Lines, Taylor, and Bagnall 2016).

From Univariate Time Series (UTS) to Multivariate Time Series (MTS), traditional methods focus on combining the compact and effective features from different dimensions, such as combined Shapelet (Cetin, Mueen, and Calhoun 2015; Grabocka, Wistuba, and Schmidt-Thieme 2016; Mousheimish, Taher, and Zeitouni 2017), global discriminative patterns (Nayak et al. 2018)), or bag-of-features (Gokce Baydogan et al. 2015; Schäfer and Leser 2017b). However, the predefined features do not always capture MTS's essentials, requiring prior knowledge of the data. Deep learning-based methods allow the end-to-end MTS modeling by various specially designed structures, showing promising performance on MTS classification task. However, whether considering the dimensional interactions (Karim et al. 2018) or not (Zheng et al. 2014; Che et al. 2018), they are mostly supervised methods, and rarely consider the shortage of MTS labels when building the classifier.

Representation learning (Bengio, Courville, and Vincent 2013) becomes a popular option when handling weakly labeled MTS, which generally learns low dimensional embeddings in an unsupervised manner, such as using triplet loss (Franceschi, Dieuleveut, and Jaggi 2019) to regularize the embedding space, then even an SVM classifier is powerful enough when a class-separable representation is learned (Wu et al. 2018). However, pure unsupervised representation learning depends mostly on the selection of the loss function. Besides, as the label information is not utilized to learn the representation (Franceschi, Dieuleveut, and Jaggi 2019), there is a risk that it deviates from the true features, thus affecting the classifier performance.

To handle both the MTS complex structure and the limited training label problem, we propose SMATE, a Semi-supervised Spatio-temporal representation learning on MultivAriate Time sEries representation. The auto-encoder based structure allows mapping the MTS samples from raw features space $\mathcal{X}$ to low dimensional embedding space

$\mathcal{H}$. A spatial modelling block combined with multi-layer convolutional network captures the spatial dynamics interactions between one-dimensional series, meanwhile, with the temporal dynamic features extracted by a GRU-based structure, SMATE is capable of compressing the essential Spatio-temporal characteristics of MTS samples into low-dimensional representations. On top of the the reconstruction objective which reduces the data dimensions, we propose a semi-supervised three-step regularization process to encourage the model in learning cluster-specific representations, where both the labeled and unlabeled samples contribute to the model's optimization.

To summarize, the main contributions of this paper are to achieve the following goals:

- **Spatio-temporal dynamic embedding for MTS:** The temporal features and the spatial interactions between MTS variables are embedded into a low-dimension space.
- **Weak supervision on learning representations:** With limited labeled data, the model can learn reliable class-separable MTS representations for subsequent learning tasks, such as building a MTS classifier.
- **Joint optimization with unlabeled samples:** The embedding space is learned via a joint optimization objective of reconstruction loss and a three-step regularization process combining both labeled and unlabeled samples.
- **Extensive experiments on the MTS datasets:** Both supervised and semi-supervised tests are conducted on various MTS datasets from different application domains.

The rest of this paper starts with a review of the most related work in the state-of-the-art. Then, we formulate the problems of the paper. Later, the proposed method on learning Spatio-temporal representation over weakly labeled MTS is presented in detail, which is followed by the experiments on real-life datasets and the conclusion.

Related Work

In this section, we discuss firstly the related work on learning MTS representation with main extension to classification task. Then, we briefly review the previous work on semi-supervised Time Series learning.

Multivariate Time Series Representation Learning

Followed the validated representations (Wang et al. 2013) on Univariate Time Series (UTS), many work extend them to MTS. For instance, authors in (Li, Li, and Fu 2016) further explored Single Value Decomposition (SVD) with multi-view learning to find the consistency and interactions between dimensions. (Cetin, Mueen, and Calhoun 2015)(Grabocka, Wistuba, and Schmidt-Thieme 2016)(Patri et al. 2015) combine local Shapelet representation from different dimensions to build an ensemble-like learner. Bag-of-Patterns representation has been also adopted for MTS. (Gokce Baydogan et al. 2015) proposed Symbolic Representation for Multivariate Time Series (SMTS), which considers all attributes of MTS simultaneously and constructs a codebook to model the local relationships. WEASEL+MUSE (Schäfer and Leser 2017b) extend WEASEL (Schäfer and Leser 2017a) from UTS to MTS by creating a histogram of feature counts to capture information about local and global changes in the MTS along different dimensions.

Different from the feature based representations, the recent end-to-end deep learning models show promising performance on both classification and regression tasks. Various network structures bring different MTS representations in the middle layer. Multi-Channels Deep Convolution Neural Networks (MC-DCNN) (Zheng et al. 2014) extract firstly 1D-CNN features from each dimension, then combine them by a Fully Connected (FC) Layer. Under similar structure, authors in (Yang et al. 2015) abandon the combination option, but apply directly 1D-CNN to all dimensions. Besides, (Che et al. 2018) described a modified GRU for modeling MTS with missing values, in which the recurrent structure allows memorizing each multivariate step into state units. A hybrid LSTM-CNN structure was firstly proposed in (Karim et al. 2017) and was further enhanced by a Squeeze-and-Excitation block (Karim et al. 2018) to model the dimensional relationship. (Hao and Cao 2020) adopt the attention mechanism to capture the dependencies on both temporal and spatial axis.

However, the works mentioned above are all supervised approaches, in which the DNN-based methods often require a massive amount of labeled data for training.

Semi-supervised Learning on Time Series

The traditional methods on Semi-supervised TS Learning are usually based on self-learning (Wei and Keogh 2006) or Positive Unlabeled Learning(Nguyen, Li, and Ng 2011)(He et al. 2015) with carefully designed distance measure, such as DTW (Wei and Keogh 2006) or DTW-D (Chen et al. 2013). Authors in (Ratanamahatana and Wanichsan 2008) propose a novel stopping criterion for optimising the semi-supervised classifier. Though not mentioned in their paper, the aforementioned self-learning framework is extensible to MTS setting by using an adapted distance measure, such as DTW_I (Shokoohi-Yekta, Wang, and Keogh 2015), DTW_D (Shokoohi-Yekta, Wang, and Keogh 2015) or DTW_A (Shokoohi-Yekta et al. 2017). However, under more complex scenarios nowadays, such as 30 MTS datasets collected from different domains in UEA archive (Anthony Bagnall and Keogh 2019), the distance-based classifiers are always considered as baselines (Bagnall et al. 2017) in recent representation based methods.

The unsupervised scalable representation learning in (Franceschi, Dieuleveut, and Jaggi 2019) combines causal dilated convolutions with unsupervised triplet loss. On the one hand, it shows a better ability for learning the UTS representation than traditional supervised convolutions (Wang, Yan, and Oates 2017). On the other hand, the combination of the MTS representation with SVM achieves better classification performance than DTW_D. Similarly, the recent proposed TapNet (Zhang et al. 2020) proposes a multi-view learning like method, Random Dimension Permutation, for weighting the contributions from each view. Extended to semi-supervised settings, Semi-TapNet achieves better performance on some training/testing imbalanced datasets. However, instead of focusing on weak-labeled training set,

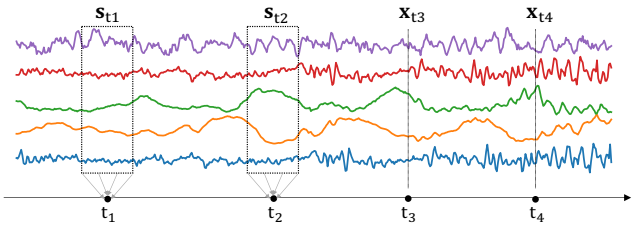

Figure 1: The spatial and temporal structure in a MTS sample representing "Walking" activity (with 5 sensors) of the SHL dataset.

Semi-TapNet utilizes unlabeled testing data to optimize the trained model, which contradicts with the usual training process and did not show the viability of the semi-supervised model.

Problem Formulation

In this section, we formulate firstly the Spatio-temporal feature learning problem in multivariate time series. Then we give a formal definition of semi-supervised classification problem for multivariate time series. The notations used in this paper are summarized in Table 1.

Table 1: Notation

Notation	Description
$\mathcal{D}, \mathcal{D}_{label}, \mathcal{D}_{unlabel}$	dataset, labeled, unlabeled portion
T, M	MTS length, dimension size
l, d	embedding length, dimension size
h	hidden representation
$\mathbf{s}$	system status / spatial interaction
θ	general parameters to be optimized

Spatio-temporal representation for MTS

Let $\mathcal{X}$ be a vector space of temporal features, $\mathcal{S}$ of spatial features, and let $\mathbf{X} \in \mathcal{X}$, $\mathbf{S} \in \mathcal{S}$ be random vectors that can be instantiated with specific values denoted $\mathbf{x}$ and $\mathbf{s}$. We can denote the training dataset $\mathcal{D} = \{\mathbf{x}_{i,1:T}\}_{i=1}^{N}$, where N is the number of MTS training samples, T is the length of MTS.

Figure 1 shows a MTS sample representing the *Walking* activity in the Sussex-Huawei Locomotion and Transportation (SHL) Dataset (Morales et al. 2017). The system status (i.e., spatial interaction) at time stamp t is not only decided by the local value $\mathbf{x}_t \in \mathcal{R}^M$ giving a temporal status, where M is the dimension size, but also by its neighbor values $\mathbf{s}_t = \left[\mathbf{x}_{t-m/2}, \mathbf{x}_{t+m/2}\right] \in \mathcal{R}^{M \times m}$, which brings a 2D spatial correlation on temporal neighbors and dimensional variables and gives a specific spatial status $\mathbf{s}_t$ for time tick t.

Giving a sample $\mathbf{x} \in \mathcal{R}^{T \times M}$, the representation learning on multivariate time series is to learn a low-dimensional embedding $\mathbf{h} \in \mathcal{R}^{l \times d}$, which integrates both temporal dynamic $p(\mathbf{x}_t|\mathbf{x}_{t'})$ and spatial dynamic $p(\mathbf{s}_t|\mathbf{s}_{t'})$ features. The item *dynamic* refers the unstable system status at different moment with evolving multivariate sequential data. Modeling the Spatio-temporal dynamics $(\mathbf{x}_t, \mathbf{s}_t) \Rightarrow (\mathbf{x}_{t'}, \mathbf{s}_{t'})$ has critical effects to the model performance (Song et al. 2018).

Semi-Supervised Learning on MTS

We believe every collected sample has its value even for unlabeled one. The sparse-label data is capable of training an effective model with the aid of unlabeled samples. The training dataset denoted as $\mathcal{D} = \{\mathcal{D}_{label}, \mathcal{D}_{unlabel}\}$, where

$$\mathcal{D}_{unlabel} = \{\hat{\mathbf{x}}_{i,1:T}\}_{i=1}^{N*(1-ratio)}, \mathcal{D}_{label} = \{\mathbf{x}_{i,1:T}, y_n\}_{i=1}^{N*ratio}$$

$ratio \in [0, 1]$ is the proportion of the labeled samples in $\mathcal{D}$. The semi-supervised MTS learning aims at training a classifier to predict successfully the label of a testing MTS sample, adopting the supervised training from $\mathcal{D}_{label}$ and further unsupervised adjustment/optimization from $\mathcal{D}_{unlabel}$.

Proposal: SMATE

This section is organised as follows. First, we show the global structure of SMATE and the main intuition of the model. Then, we describe how the Spatio-temporal representation is learned from the raw MTS data space. Finally, we give the joint optimization of the model which coordinates the weak supervision and the embedding learning via a three-step regularization process.

Global Structure of SMATE

SMATE is based on an asymmetric auto-encoder structure, which contains three key components: Spatio-temporal dynamic encoder, sequential decoder, and semi-supervised regularization on embedding space.

Giving the fact that extracting features from high-dimension space generally requires additional attention compared to restoring data from low-dimension space (Bengio, Courville, and Vincent 2013), the encoding and decoding process in SMATE adopts different weight matrix to better capture the inner structure of MTS data. Although recent work (Ma et al. 2019) do represent the temporal dynamic of MTS via sequence to sequence (seq2seq) model, they do not actually encompass the complex Spatio-temporal structure of MTS. As shown in Figure 2, SMATE adopts a two-channel encoder working on both spatial and temporal feature extraction. In multivariate time series, the spatial features generally refer to the spatial interactions between each 1-D series, which can be captured by the spatial modelling block. The Spatio-temporal encoder embeds the input MTS samples into a low dimensional representation space, where the embedded samples are sparsely distributed with the reconstruction-based optimisation. On top of the unsupervised embedding space, a three-step regularization process is applied to learn a class-specific Spatio-temporal representation, where the class centroids are regularized by both the labeled and unlabeled samples. A joint optimization of the reconstruction and regularization objectives allows improving the reliability of the learned representation with a weak supervision.

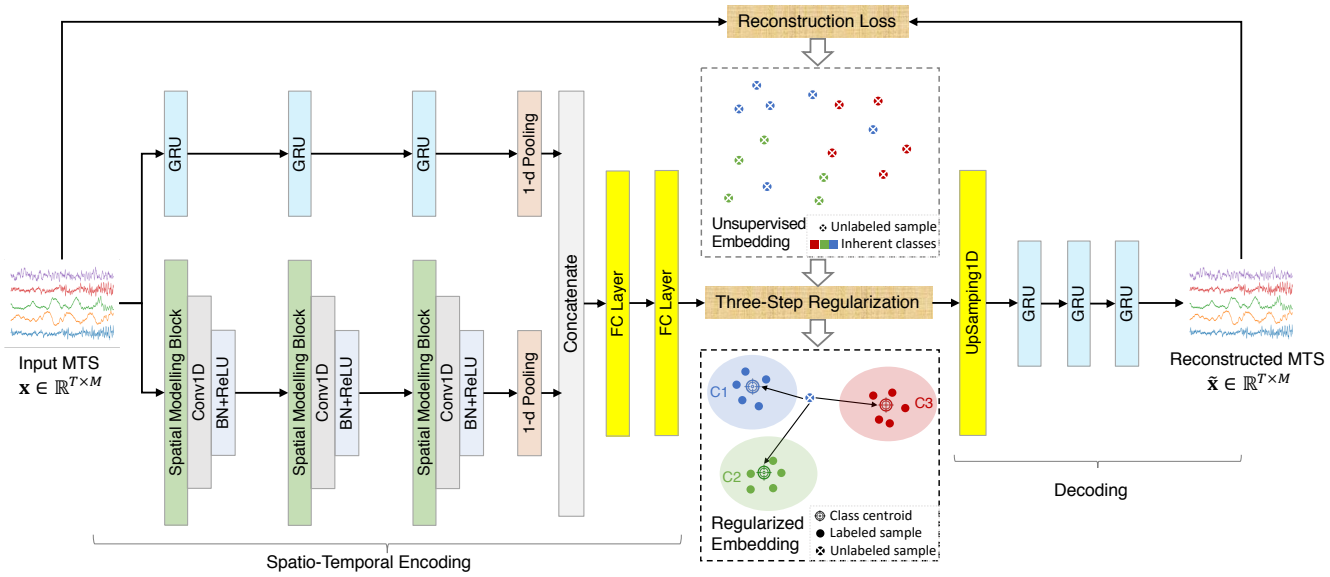

Figure 2: Model Structure of SMATE

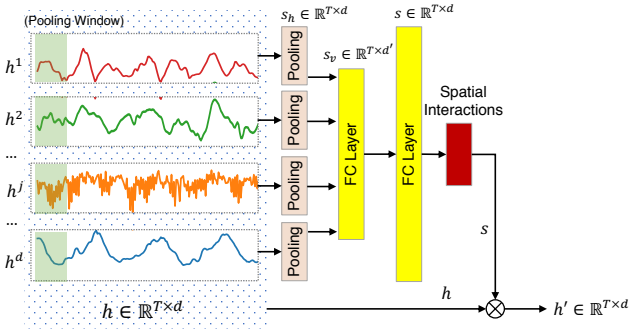

Figure 3: The Spatial Modelling Block

Spatial Modelling Block

We firstly introduce a novel module, Spatial Modelling Block (SMB), for capturing the dimensional interactions (i.e., system status) between 1-D series. As shown in Figure 3, SMB takes as input a multivariate time series representation $h \in \mathcal{R}^{T\times d}$ ($x \in \mathcal{R}^{T\times M}$ for the first block in spatial encoding channel), a one-dimensional average pooling layer is then applied to each MTS dimension, encoding the temporal neighbors into the horizontal system status $s_{h_i} = avg([h_{i-m/2}, h_{i+m/2}])$, where i is the time tick, m is the pool size. Then the Fully Connected (FC) layers allow firstly interacting the horizontal system status s_h in vertical direction via a low-dimension compression $s_v \in \mathcal{R}^{T\times d'}$, then remapping it to the initial data space to decide the spatial interaction weights at each one-dimensional segment. We define the spatial interactions $\mathbf{s} = \{s_i\}_{i=1}^{T}$, where $s_i \in \mathcal{R}^d$, representing the interaction weights for the vector h_i. The output of SMB is described by $h' = h \bigodot \mathbf{s}$, with the calibrated weights for each 1-D TS segment. The pooling window matches exactly to the kernel size in the concatenated 1-D convolutional network, as the convolution product $K \bigodot [h_{i-m/2}, h_{i+m/2}]$ requires considering the spatial interactions captured by SMB within the same interval.

Spatio-Temporal Encoding on MTS

Given $\mathbf{x} \in \mathcal{R}^{T\times M}$ as a MTS sample, the objective of the multivariate time series encoding is to provide a low-dimension representation $\mathbf{h} \in \mathcal{R}^{l\times d}$ in the embedding space $\mathcal{H}$, which compresses the Spatio-temporal features of $\mathbf{x}$ from the raw data space $\mathcal{X}$ by a neural network based function $f_\theta(\mathbf{x})$. The low-dimensional embedding has been proven effective for classification task (Zhang et al. 2020), with a dramatic improvement on both the efficiency and accuracy of the prediction, owning to the fact that the classifier is not distracted by the redundant information in raw data.

As aforementioned, we adopt a two-channel structure to encode respectively the spatial and temporal dynamic features in multivariate time series. For the temporal channel, we apply Gated Recurrent Units (GRUs) (Chung et al. 2014) to encode the temporal dynamics. Each observation $\mathbf{x}_t \in \mathcal{R}^M$ is compressed into the cell value $\tilde{h}_t$, which combines with selected previous memory to output the hidden value $h_t \in \mathcal{R}^{d_g}$, where d_g is the hidden dimension of GRU with the default value 128. Then an 1-D pooling layer is applied to get the output $h_{gru+pool} \in \mathcal{R}^{l\times d_g}$, where $l = T/pool\_size$. For the spatial channel, three convolutional modules are applied. Within each module, the Spatial Modelling Block (SMB) firstly calibrates the interaction weights for each 1-D segments and outputs $h' \in \mathcal{R}^{T\times d}$. Then an 1-D convolutional layer concatenated with Batch Normalization (Ioffe and Szegedy 2015) and Rectified Linear Units (ReLU) (Nair and Hinton 2010) is deployed. We set the default value of filters for the three convolutional layers as 128, 256, 128 and the kernels as 8, 5 and 3. The kernels match to

the pool size in their neighboring SMB. The output of each convolutional module is $h_{conv} \in \mathcal{R}^{T \times d_c}$, where d_c is the filter size of each convolutional layer. Similar as the temporal channel, an 1-D pooling layer is applied after the last convolutional block to get the output $h_{cnn+pool} \in \mathcal{R}^{l \times d_c}$. Then, to combine the extracted spatial and temporal features, we concatenate the results of the two channels and get the output $h_{concat} \in \mathcal{R}^{l \times (d_l + d_c)}$. Last, we apply two fully connected layers on h_{concat} to get the low-dimensional representation $h \in \mathcal{R}^{l \times d}$. The detailed parameter settings can be found in the Appendix. The two dimensional representation allows maximizing the contention of the Spatio-temporal features, and facilitating the MTS restoration.

Joint Model Optimisation

As shown in Figure 2, since the representation learned via an auto-encoder based structure generally has a sparse distribution of class-specific samples (Bengio, Courville, and Vincent 2013), the unsupervised training derived from the reconstruction objective does not consider thoroughly the inner reliance between class-specific samples. Therefore, we propose a joint model optimization which integrates the temporal reconstruction and the three-step regularization objectives combining both labeled and unlabeled samples.

Firstly, the *reconstruction loss* is defined as:

$$L_R = \mathbb{E}_{\mathbf{x}_{1:T}} \big[\sum_t \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2 \big] \tag{1}$$

where $\mathbf{x}_t$ and $\tilde{\mathbf{x}}_t$ correspond respectively the initial and reconstructed vector at time t.

Then, we introduce the three-step regularization objective which combines both labeled and unlabeled samples to train the distance-based loss function. Similar as the Prototypical Networks (Snell, Swersky, and Zemel 2017), but considering the label shortage, the regularization allows the embedded samples within the class-specific clusters to approach the virtual class centroids which are trained progressively.

Supervised Centroids Initialization The class centroids are initialized by class-specific embeddings. Giving the labeled samples $\mathcal{D}_{label} = \{X^k\}_{k=1}^K$ where K is the number of classes in the labeled training set, $X^k \in \mathcal{R}^{N_k \times T \times M}$ is a sample collection of class k. The embedding collection $H^k \in \mathcal{R}^{N_k \times l \times d}$ then initializes the centroid of class k by:

$$c_k = mean(H^k), \quad c_k \in \mathcal{R}^{l \times d} \tag{2}$$

Supervised Centroids Adjustment Once the centroids are initialized, the supervised adjustment can be made owning to the fact that distance-based class probability allows to assess the contribution of individual samples on centroid's decision. The centroid is affected with larger contribution weights by near-by samples. We define the weight of $\mathbf{x}_i \in \mathcal{R}^{T \times M}$ to c_k as the inverse euclidean distance between $h_i = f_\theta(\mathbf{x}_i)$ and the centroid c_k:

$$W_{k,i} = 1 - \frac{dist(h_\theta(\mathbf{x}_i), c_k)}{\sum_{j=1}^K dist(h_\theta(\mathbf{x}_i), c_j)} \tag{3}$$

Then the class centroid can be adjusted accordingly by the labeled samples within the class-specific cluster:

$$c_k = \sum_{i=1}^{N_k} W_{k,i} \cdot h_i^k, \quad h_i^k \in H^k \tag{4}$$

Unsupervised Centroids Adjustment Giving the unlabeled samples $\mathcal{D}_{unlabel} = \{\hat{\mathbf{x}}_i\}_{i=1}^{N*(1-ratio)}$, where $ratio$ is the labeled data proportion. Apart from the optimization from the reconstruction objective, the unlabeled sample $\hat{\mathbf{x}}_i$ is capable of adjusting the centroid c_k via the propagated label from the distance-based class probability defined as:

$$\hat{p}_\theta(y = k|\hat{\mathbf{x}}_i) = 1 - \frac{dist(h_\theta(\hat{\mathbf{x}}_i), c_k)}{\sum_{j=1}^K dist(h_\theta(\hat{\mathbf{x}}_i), c_j)} \tag{5}$$

The unlabeled sample $\tilde{\mathbf{x}}$ will be integrated into the class-specific cluster with highest probability. The class centroid c_k is further adjusted by considering the unlabeled samples:

$$c_k = \frac{N_k}{N_k + \hat{N}_k} \sum_{i=1}^{N_k} W_{k,i} \cdot h_i^k + \frac{\hat{N}_k}{N_k + \hat{N}_k} \sum_{i=1}^{\hat{N}_k} \hat{p}_{k,i} \cdot \hat{h}_i^k \tag{6}$$

The regularization loss derived from labeled samples but with the semi-supervised centroids, is formalized as follows:

$$L_{Reg}(\theta) = -\sum_k log W_\theta(y = k|\mathbf{x}) \tag{7}$$

As both the reconstruction and regularization losses are normalized, then the global objective is defined as:

$$\min_\theta (L_R + \lambda L_{Reg}) \tag{8}$$

where $\lambda \geq 0$ is a hyperparameter that balances the two losses. Importantly, L_{Reg} is included such that the embedding process not only serves to reduce the dimensions – it is actively conditioned to facilitate the encoder in learning class-specific data clusters. In practice, SMATE was not sensitive to λ; then for all the experiments we set $\lambda = 1$.

Experiments

In this section, we evaluate the performance of the Spatio-temporal representation learned by SMATE. Firstly, we show the experimental design including the datasets information, baseline descriptions and evaluation metrics. Then we evaluate the performance of the model with different baselines on both supervised and semi-supervised learning tasks. Finally, we analyse the spatial modelling block regarding its ability of modelling the dimensional interactions in MTS. The model was trained using the Adam optimizer (Kingma and Lei Ba 2015) on a single Tesla V100 GPU of 32 Go memory with CUDA 10.2. The authors are devoted to promote reproducibility, therefore the source code, datasets and instructions will be made publicly available after the paper is accepted (The reviewers are invited to check those information in the supplementary materials).

Table 2: Performance Comparison for MTS classification over UEA MTS archive

Dataset	SMATE	USRL	TapNet	MLSTM -FCN	WEASEL +MUSE	1NN-ED	1NN-DTW$_I$	1NN-DTW$_D$	1NN-ED (norm)	1NN-DTW$_I$ (norm)	1NN-DTW$_D$ (norm)	1NN-DTW$_A$ (norm)
ArticularyWordRecognition	**0.993**	0.973	0.987	0.973	0.99	0.97	0.98	0.987	0.97	0.98	0.987	0.987
AtrialFibrillation	0.133	0.133	**0.333**	0.267	**0.333**	0.267	0.267	0.2	0.267	0.267	0.22	0.267
BasicMotions	**1**	**1**	**1**	0.95	**1**	0.675	**1**	0.975	0.676	**1**	0.975	**1**
CharacterTrajectories	0.984	0.994	**0.997**	0.985	0.99	0.964	0.969	0.99	0.964	0.969	0.989	0.989
Cricket	0.972	0.986	0.958	0.917	**1**	0.944	0.986	**1**	0.944	0.986	**1**	**1**
Epilepsy	0.964	0.957	0.971	0.761	**1**	0.667	0.978	0.964	0.666	0.978	0.964	0.979
ERing	**0.97**	0.88	0.904	0.941	0.964	0.93	0.93	0.93	0.93	0.93	0.93	0.93
EthanolConcentration	0.373	0.236	0.323	0.373	**0.43**	0.293	0.304	0.323	0.293	N/A	0.323	0.316
FaceDetection	**0.563**	0.528	0.556	0.545	0.545	0.519	0.513	0.529	0.519	0.5	0.529	0.529
FingerMovements	**0.59**	0.54	0.53	0.58	0.49	0.55	0.52	0.53	0.55	0.52	0.53	0.509
HandMovementDirection	**0.527**	0.27	0.378	0.365	0.365	0.279	0.306	0.231	0.278	0.306	0.231	0.224
Heartbeat	0.727	0.688	**0.751**	0.663	0.727	0.62	0.659	0.717	0.619	0.658	0.717	0.571
LSST	**0.591**	0.558	0.568	0.373	0.59	0.456	0.575	0.551	0.456	0.575	0.551	0.551
MotorImagery	**0.59**	0.54	**0.59**	0.51	0.5	0.51	0.39	0.5	0.51	N/A	0.5	0.5
N/ATOPS	0.883	**0.944**	0.939	0.889	0.87	0.86	0.85	0.883	0.85	0.85	0.883	0.883
PEMS-SF	**0.763**	0.688	0.751	0.699	N/A	0.705	0.734	0.711	0.705	0.734	0.711	0.73
PenDigits	0.98	**0.983**	0.98	0.978	0.948	0.973	0.939	0.977	0.973	0.939	0.977	0.977
Phoneme	0.177	**0.246**	0.175	0.11	0.19	0.104	0.151	0.151	0.104	0.151	0.151	0.151
SelfRegulationSCP1	**0.887**	0.771	0.739	0.874	0.71	0.771	0.765	0.775	0.771	0.765	0.775	0.786
SelfRegulationSCP2	**0.556**	**0.556**	0.55	0.472	0.46	0.483	0.533	0.539	0.483	0.533	0.539	0.539
SpokenArabicDigits	0.982	0.956	0.983	**0.99**	0.982	0.967	0.96	0.963	0.967	0.959	0.963	0.963
StandWalkJump	**0.533**	0.4	0.4	0.067	0.333	0.2	0.333	0.2	0.2	0.333	0.2	0.333
Avg. Rank	**2.77**	6	3.63	6.32	4.18	7.95	6.73	5.55	8.23	6.5	5.55	5.32
Wins/Ties	**12**	5	5	1	5	0	1	1	0	1	1	2

Experimental design

We evaluate the learned MTS representation on both classification and semi-supervised classification tasks. As SMATE allows learning class-separable representations, then learning a simple SVM on these features is a good way (Wu et al. 2018) to validate the representation model. We firstly train supervised representation model using the labeled training set, then we train a SVM with radial basis function kernel to output the classification score on the testing set. For semi-supervised aspect, we apply different portions of labels in training set to train the semi-supervised representation model, serving to learn the SVM classifier with the propagated labels from the distance-based class probability.

Datasets description We evaluate our proposed method on the newly released UEA archive (Anthony Bagnall and Keogh 2019). We select 22 MTS datasets from various application domains [1], which have a big difference on dimension size ($2 \sim 963$), sample length ($8 \sim 3000$) and the number of training samples ($12 \sim 7494$). The paper (Hoang et al. 2018) released with the archive shows the main application domains and the number of matching datasets in the archive are: Human Activity Recognition (9), Mothion Classification (4), ECG classification (3), EEG/MEG classification (6), Audio Spectra Classification (5), others (3).

Evaluation metrics We use the accuracy as the unique metric for supervised task, which is the default criterion in Time Series Classification work. For semi-supervised task, we evaluate the classifier's accuracy at different supervision level by varying the labeled samples in the training set.

[1]Considering the computation cost, we exclude the datasets with extremely high dimension size and long length. The dataset details can be found in www.timeseriesclassification.com

Classification Performance Evaluation

Comparison Methods We compare the performance on classification task with 11 different benchmark approaches, including both the classical data mining and emerging deep learning methods, which are summarized as follows:

- Distance-based Nearest Neighbor classifier on (non-) normalized MTS (Shokoohi-Yekta, Wang, and Keogh 2015). **1NN-ED**: Euclidean Distance ; **1NN-DTW**$_I$: Sum of Dynamic Time Warping distance on one-dimensional series; **1NN-DTW**$_D$: DTW distance on multi-dimensional vectors; **1NN-DTW**$_A$: Adaptive distance selected between DTW$_I$ and DTW$_D$ with higher accuracy at run time.
- Bag-of-patterns classifier. **WEASEL+MUSE** (Schäfer and Leser 2017b): the logistic regression classifier on top of the bag of discriminative features.
- Deep Learning based classifier. **MLSTM-FCN** (Karim et al. 2018): a multi-layer perceptron (MLP) with softmax function over the concatenated LSTM and CNN layers, focusing on the global relation between dimensions; **TapNet** (Zhang et al. 2020): distance-based classifier over embeddings, focusing on weighting different views of MTS via a set of grouped dimensions; **USRL** (Franceschi, Dieuleveut, and Jaggi 2019): SVM classifier on the representation learned via unsupervised temporal encoding.

Results Analysis Table 2 shows the accuracy results comparison between our proposition and the 11 baselines mentioned above. We show as well the average rank and the number of Wins/Ties of each method. "N/A" indicates the model is not applicable due to memory overflow. Overall,

SMATE defends its reliability with 12 Wins/Ties and the highest average rank of 2.77 among all the baselines. The current state-of-the-art deep learning method (TapNet) and the powerful data mining method (WEASEL+MUSE) have close rank (3.63/4.18). WEASEL+MUSE performs among the best in Human Activity Recognition task (*BasicMotions*, *Criket*, *Epilepsy*, *ERring*), as the class-discriminative patterns can be directly extracted from the raw data space. Besides, the unsupervised representation learning method (USRL) performs much worse than SMATE with the same SVM classifier, which confirms the reliability of the semi-supervised regularization in the embedding space. Moreover, SMATE achieves the best performance among the baselines on all the datasets of EEG/MEG applications (*FaceDetection*, *FingerMovements*, *HandMovementDirection*, *MotorImagery*, *SelfRegulationSCP1*, *SelfRegulationSCP2*), where the 1-D series (i.e., signals) generally have strong dependencies with each others. The dimensional interactions could be essential characteristics that SMATE has successfully captured.

However, SMATE produces visibly low accuracy on some datasets, for instance, 0.133 on *AtrialFibrillation*, 0.177 on *Phoneme*, on which the baselines perform poorly as well. This is probably caused by the original data source.

Semi-supervised Classification Performance

We select four representative datasets from different application domains to validate the semi-supervised aspect of SMATE. Two recently proposed semi-supervised models: **USRL** (Franceschi, Dieuleveut, and Jaggi 2019) and **Semi-TapNet** (Zhang et al. 2020) are adopted for the comparison. While Semi-TapNet declares its viability with the controversial optimization via unlabeled testing data, we prefer learning the model only with the training set of different supervision ratios, which is more realistic in practical scenarios.

Table 3 shows the classification accuracy at different supervision level. In Motion Recognition task, from 10% labeled training set to fully labeled one, the accuracy of SMATE varies only by 0.046, compared to USRL (0.286) and Semi-TapNet (0.151), showing that SMATE is capable of learning a class-separable representation with a weak supervision, which shows better classification performances than other classifiers with the intense supervision. For instance, on *SelfRegulationSCP1*, with 10% labeled samples, SMATE is capable of obtaining a higher accuracy (0.781) than fully supervised USRL (0.771) and TapNet (0.739).

Performance of Spatial Modelling Block (SMB)

To validate the Spatial Modelling Block (SMB), we firstly compare the classification accuracy of SMATE with or without integrating SMB. Then we re-build SMATE by replacing SMB with the following modules in the state-of-the-art work which learn the dimension relationships of MTS: **Random Dimension Permutation (RDP)** in TapNet (Zhang et al. 2020) and **Squeez-and-Excitation (SE)** in MLSTM-FCN (Karim et al. 2018). Briefly, SMATE-SMB achieves [14 Wins|6 Ties|2 Losses] to SMATE-NonSMB, thus indicating SMB does contribute to a better representation.

Table 3: Semi-supervised performance on different domains

Dataset (Domain)	Method	ratio (0.1)	ratio (0.2)	ratio (0.5)	ratio (1.0)
ArticularyWordRec. (Motion)	USRL	0.686	0.843	0.942	0.973
	Semi-TapNet	0.836	0.858	0.949	0.987
	SMATE	0.947	0.977	0.987	0.993
Epilepsy (Human Activity)	USRL	0.675	0.891	0.93	0.957
	Semi-TapNet	0.547	0.606	0.921	0.971
	SMATE	0.739	0.92	0.957	0.964
Heartbeat (Audio Spectra)	USRL	0.70	0.706	0.703	0.688
	Semi-TapNet	0.623	0.644	0.712	0.751
	SMATE	0.717	0.702	0.722	0.727
SelfRegulationSCP1 (EEG/MEG)	USRL	0.646	0.732	0.76	0.771
	Semi-TapNet	0.646	0.668	0.706	0.739
	SMATE	0.781	0.836	0.867	0.887

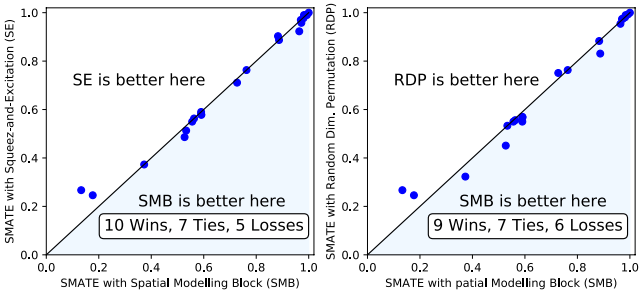

Figure 4: Accuracy comparison between SMB and SE/RDP

In Figure 4, we give a one-to-one comparison between SMB and SE/RDP over all aforementioned datasets. While there is no distinct accuracy difference, we find that SMATE performs better than other modules on modeling the dimensional interactions: [10 Wins|7 Ties|5 Losses] copared to SE, [9 Wins|7 Ties|6 Losses] to RDP. RDP performs relatively better than SE, as a set of grouped dimensions provides various views of MTS, which may allow exploring the interactions between the subsets of all variates more thoroughly. However, extra parameters for grouping the dimensions are introduced. SE is a parameter-free module but considers each variate has a unique and stable state when interacting with other variates, which ignores the dynamic features of time series data. SMB answers both the questions of the parameter-free settings and the dynamic interactions. The results show that capturing the spatial dynamic interactions at the sub-sequence level performs better than modeling the dimensional interactions at the sequence level.

Conclusion

In this paper, we proposed SMATE, to learn the Spatial-temporal representation on weakly-labeled multivariate time series. Inside the auto-encoder based structure, the Spatial-temporal encoder maps temporal dynamic features and the spatial dynamic interactions between 1-D series into a low dimensional embedding space. A three-step regularization process is proposed to encourage the model in learning class-separable representation, where a weak supervision allows building a reliable classifier. The learned model is validated on 22 real-life datasets in UEA MTS archive, with

eleven baselines for supervised classification task and two baselines for semi-supervised learning task.

References

Anthony Bagnall, Jason Lines, W. V.; and Keogh, E. 2019. The UEA & UCR Time Series Classification Repository. www.timeseriesclassification.com, Date of Access: 2020/05.

Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery, 2017* .

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8): 1798–1828.

Cetin, M. S.; Mueen, A.; and Calhoun, V. D. 2015. Shapelet ensemble for multi-dimensional time series. In *SIAM International Conference on Data Mining 2015, SDM 2015*.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values 8: 6085. doi:10.1038/s41598-018-24271-9. URL www.nature.com/scientificreports.

Chen, Y.; Hu, B.; Keogh, E.; and Batista, G. E. A. P. A. 2013. DTW-D: Time Series Semi-Supervised Learning from a Single Example. In *KDD'13*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *NeurIPS'19*.

Gokce Baydogan, M.; Runger, G.; Baydogan, M. G.; and Runger, G. 2015. Learning a symbolic representation for multivariate time series classification. *Data Min Knowl Disc* 29: 400–422.

Grabocka, J.; Wistuba, M.; and Schmidt-Thieme, L. 2016. Fast classification of univariate and multivariate time series through shapelet discovery. *Knowledge and Information Systems* 49(2): 429–454.

Hao, Y.; and Cao, H. 2020. A New Attention Mechanism to Classify Multivariate Time Series. In *IJCAI'20*.

He, G.; Duan, Y.; Li, Y.; Qian, T.; He, J.; and Jia, X. 2015. Active Learning for Multivariate Time Series Classification with Positive Unlabeled Data doi:10.1109/ICTAI.2015.38.

Hoang, A. B.; Dau, A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; and Keogh, E. 2018. The UEA multivariate time series classification archive, 2018. Technical report. URL www.timeseriesclassification.com.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*.

Karim, F.; Majumdar, S.; Darabi, H.; and Chen, S. 2017. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* 6: 1662–1669.

Karim, F.; Majumdar, S.; Darabi, H.; and Harford, S. 2018. Multivariate LSTM-FCNs for Time Series Classification. Technical report.

Kingma, D. P.; and Lei Ba, J. 2015. Adam: A Method for Stochastic Optimization. Technical report.

Li, S.; Li, Y.; and Fu, Y. 2016. Multi-View Time Series Classification: A Discriminative Bilinear Projection Approach .

Lines, J.; Taylor, S.; and Bagnall, A. 2016. HIVE-COTE: The Hierarchical Vote Collective of Transformation-based Ensembles for Time Series Classification. In *IEEE ICDM'16*.

Ma, Q.; Zheng, J.; Li, S.; and Cottrell, G. W. 2019. Learning Representations for Time Series Clustering. In *Advances in Neural Information Processing Systems (NeurIPS'19)*.

Morales, O.; Javier, F.; Gjoreski, H.; Ciliberto, M.; Wang, L.; Javier Ordonez Morales, F.; Mekki, S.; Valentin, S.; Member, S.; and Roggen, D. 2017. The University of SussexHuawei locomotion and transportation dataset for multimodal analytics with mobile devices The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics with Mobile Devices .

Mousheimish, R.; Taher, Y.; and Zeitouni, K. 2017. Automatic Learning of Predictive CEP Rules: Bridging the Gap between Data Mining and Complex Event Processing. In *DEBS '17*.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML '10*.

Nayak, G.; Mithal, V.; Jia, X.; and Kumar, V. 2018. Classifying multivariate time series by learning sequence-level discriminative patterns. In *SIAM International Conference on Data Mining, SDM 2018*, 252–260.

Nguyen, M. N.; Li, X. L.; and Ng, S. K. 2011. Positive unlabeled learning for time series classification. In *IJCAI International Joint Conference on Artificial Intelligence*.

Patri, O. P.; Kannan, R.; Panangadan, A. V.; and Prasanna, V. K. 2015. Multivariate Time Series Classification Using Inter-leaved Shapelets. *Time Series Workshop @ NIPS 2015* 1–5.

Ratanamahatana, C. A.; and Wanichsan, D. 2008. Stopping Criterion Selection for Efficient Semi-supervised Time Series Classification. *Soft. Eng., Arti. Intel., Net. Para./Distri. Comp.* .

Schäfer, P.; and Leser, U. 2017a. Fast and Accurate Time Series Classification with WEASEL. In *CIKM'17*.

Schäfer, P.; and Leser, U. 2017b. Multivariate Time Series Classification with WEASEL+MUSE. Technical report.

Shokoohi-Yekta, M.; Hu, B.; Jin, H.; Wang, J.; and Keogh, E. 2017. Generalizing DTW to the multi-dimensional case requires an adaptive approach HHS Public Access. *Data Min Knowl Discov* 31(1): 1–31.

Shokoohi-Yekta, M.; Wang, J.; and Keogh, E. 2015. On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case. In *SDM'15*.

Snell, J.; Swersky, K.; and Zemel, T. R. 2017. Prototypical Networks for Few-shot Learning. In *NIPS'17*.

Song, D.; Xia, N.; Cheng, W.; Chen, H.; and Tao, D. 2018. Deep r -th Root of Rank Supervised Joint Binary Embedding for Multivariate Time Series Retrieval. *KDD '18* .

Tang, W.; Long, G.; Liu, L.; Zhou, T.; Jiang, J.; and Blumenstein, M. 2020. Rethinking 1D-CNN for Time Series Classification: A Stronger Baseline. Technical report.

Wang, X.; Mueen, A.; Ding, H.; Trajcevski, G.; Scheuermann, P.; and Keogh, E. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery* 26(2): 275–309.

Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2017-May, 1578–1585.

Wei, L.; and Keogh, E. 2006. Semi-supervised time series classification. In *Proc. ACM SIGKDD'06*.

Wu, L.; En-Hsu, I.; Yi, Y. J.; Xu, F.; Lei, Q.; and Witbrock, M. J. 2018. Random Warping Series: A Random Features Method for Time-Series Embedding. In *AISTATS'18*.

Yang, J.; Nguyen, M. N.; San, P. P.; Li, X. L.; and Krishnaswamy, S. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *IJCAI'15*.

Zhang, X.; Gao, Y.; Lin, J.; and Lu, C.-T. 2020. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. In *AAAI'20*.

Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; and Zhao, J. L. 2014. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In *WAIM'14*.

SMATE: Semi-Supervised Spatio-Temporal Representation Learning on Multivariate Time Series (Appendix)

Jingwei Zuo, Karine Zeitouni and Yehia Taher
DAVID Lab, University of Versailles, Université Paris-Saclay, Versailles, France
Email: {jingwei.zuo, karine.zeitouni, yehia.taher}@uvsq.fr

Experimental Setup

In this section, we supplement detailed experiment setups for the tasks.

Dataset Descriptions

Considering the computation cost, we exclude the datasets with extremely high dimension size and long length of UEA MTS archive. We take 22 datasets from different domains to validate the universality of SMATE. The datasets information can be found in Table 5. We adopt the default train/test split of the archive.

Hyper-parameters Setup

Network Architecture As shown in Table 4, the Spatio-temporal encoder is composed by the Temporal Channel, Spatial Channel and Fully Connected (FC) layers. In Temporal Channel, we denote the Gated-Recurrent-Unit layer as *GRU*, in brackets we give the hidden dimension size of GRU. An one-dimensional Average Pooling layer is concatenated with GRUs, we give the pool size, stride and padding in the bracket; The Spatial Channel contains 3 convolutional blocks and one identical pooling layer as the Temporal Channel. Each convolutional block is composed by a Spatial Modelling Block (*SMB*), an 1-D convolutional layer (*Conv1D*), a batch normalization layer (*Batch Norm*) and a ReLu activation layer. In brackets of *Conv1D*, we provide the kernel size, stride and padding, which are followed by a dash with the number of filters. The SMBs are configured with consistent parameters of its neighbor *Conv1D*; The decoder contains one UpSampling AD layer and 3 GRU layers. The default values are shown in Table 4.

However, as the datasets are collected from different domains with big difference on dimension size $M \in [2, 963]$, sample length $T \in [8, 3000]$ and sample numbers $N \in [12, 7494]$, it's unpractical to apply a unified parameter setting on all datasets. In particular, the data dimension size M have huge impact to the hidden dimension size $d^{'}$ of FC layer in SMB_1. Similarly, the sample length T affects the pool size $\mathcal{P}$ in the pooling layers and UpSampling1D layer. We provide the detailed parameter settings for each datasets in Table 5. The convolutional kernels for *PenDigits* dataset are set to (4,1,1) as it is infeasible to apply our default kernel size "8" into a 8-length time series in "PenDigits" dataset.

Table 4: Network Architecture of SMATE

Module	Layer	Type
Temporal Channel	1	GRU (128)
	2	GRU (128)
	3	GRU (128)
	4	AveragePooling1D($\mathcal{P}$, $\mathcal{P}$, 0)
Spatial Channel	1	SMB_1 + Conv1D(8,1,0) -128 filters + Batch Norm + ReLU
	2	SMB_2 + Conv1D(5,1,0) -256 filters + Batch Norm + ReLU
	3	SMB_3 + Conv1D(3,1,0) -128 filters + Batch Norm + ReLU
	4	AveragePooling1D($\mathcal{P}$, $\mathcal{P}$, 0)
FC	1	FC (128) + Batch Norm + LeakyReLU
	2	FC (128) + Batch Norm
SMB_1	1	AveragePooling1D(8, 1, 0)
	2	FC ($d^{'}$) + ReLU
	3	FC (M) + Sigmoid
SMB_2	1	AveragePooling1D (5, 1, 0)
	2	FC (8) + ReLU
	3	FC (128) + Sigmoid
SMB_3	1	AveragePooling1D (3, 1, 0)
	2	FC (16) + ReLU
	3	FC (256) + Sigmoid
Decoder	1	UpSampling1D ($\mathcal{P}$)
	2	GRU (128)
	3	GRU (128)
	4	GRU (M)

Experiment Parameters The Adam optimizer is adopted for model's training, with a learning rate of 0.00001 and the default exponential decay rate in Keras. As there are limited training samples in most of the UEA datasets, it is not feasible to separate a validation set from the small size of training samples. For instance, the dataset "StandWalkJump" contains only 12 training samples for 3 classes. It is unpractical to split the small training samples into training and validation sets. Therefore, we define the stop condition only based on the training loss. We set the stop condition to hold when the difference of training loss between epochs is less than a small threshold, 0.0001 for 3 consecutive steps.

Table 5: MTS dataset information & Training Parameter Settings

Domain	Dataset	Samples	Dim. (M)	Length (T)	Class	$\mathcal{P}$	d'
Human Activity	BasicMotions	40/40	6	100	4	10	2
	Cricket	108/72	6	1197	12	100	4
	Epilepsy	137/138	3	206	4	30	2
	ERing	30/270	4	65	6	5	4
	N/ATOPS	180/180	24	51	6	3	6
Motion	ArticularyWordRecognition	275/300	9	144	25	10	4
	CharacterTrajectories	1422/1436	3	182	20	30	2
	PenDigits	7494/3498	2	8	10	2	2
ECG	AtrialFibrillation	15/15	2	640	3	64	2
	StandWalkJump	12/15	4	2500	3	100	4
EEG/MEG	FaceDetection	5890/3524	144	62	2	8	16
	FingerMovements	316/100	28	50	2	5	8
	HandMovementDirection	160/74	10	400	4	40	4
	MotorImagery	278/100	64	3000	2	100	8
	SelfRegulationSCP1	268/293	6	896	2	100	6
	SelfRegulationSCP2	200/180	7	1152	2	100	6
Audio Spectra	Heartbeat	204/205	61	405	2	5	16
	Phoneme	3315/3353	11	217	39	30	4
	SpokenArabicDigits	6599/2199	13	93	10	10	4
Others	EthanolConcent.	261/263	3	1751	4	50	2
	LSST	2459/2466	6	36	14	4	4
	PEMS-SF	267/173	963	144	7	10	64

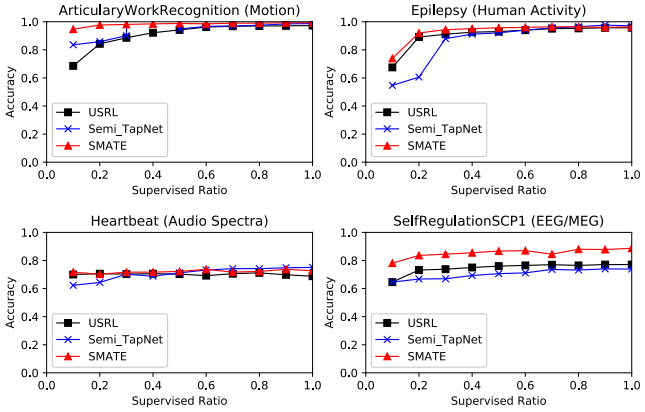

Figure 5: Semi-supervised performance on four datasets from different domains

Baselines Selection

We are not capable of including all (semi-)MTSC work into the comparison. Some MTSC work like MC-DCNN (Zheng et al. 2014) has been beaten by our baselines in their papers, which is not necessary to be re-compared. However, we include the 1-NN based classifiers with different distance measures, as they are considered as the benchmarks in MTSC work by the community. The semi-supervised MTSC approach proposed in (Wei and Keogh 2006) is not included in the comparison neither, as the 1-NN based classifier adopted in the paper shows no advantage nowadays compared to the advanced approaches in Table 2. From our experiments, the fully supervised 1-NN classifier can not even beat our semi-supervised SMATE. We show a detailed semi-supervised results of Table 3 in Figure 5.

Advantages Discussion

In this section, we discuss the advantages of SMATE over existing supervised and semi-supervised MTSC. First, owing to the Spatio-Temporal dynamic encoder, SMATE allows explore more thoroughly the essential characteristics of MTS. TapNet (Zhang et al. 2020) and MLSTM-FCN (Karim et al. 2018) consider generally the correlation between the entire 1-D series, while USRL processes indifferently the MTS and UTS (Franceschi, Dieuleveut, and Jaggi 2019), they all ignore the fact that the interactions between 1-D segments may evolve due to the dynamic features in the sequential data, which reflect the key MTS characteristics in certain domains (e.g., EEG/MEG applications). Second, SMATE explores thoroughly the valuable information from the unlabeled samples, which contributes not only to the auto-encoder's reconstruction objective, but also to the regularization process on the embedding space. While Semi-TapNet (Zhang et al. 2020) considers unlabeled data only with the pseudo labels predicted by intermediate-trained classifier, which is not reliable when there are limited labeled samples for the classifier's training. USRL (Franceschi, Dieuleveut, and Jaggi 2019) trains the representation without any supervision, which has no advantage in classification tasks.

References

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *NeurIPS'19*.

Karim, F.; Majumdar, S.; Darabi, H.; and Harford, S. 2018. Multivariate LSTM-FCNs for Time Series Classification. Technical report.

Wei, L.; and Keogh, E. 2006. Semi-supervised time series classification. In *Proc. ACM SIGKDD'06*.

Zhang, X.; Gao, Y.; Lin, J.; and Lu, C.-T. 2020. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. In *AAAI'20*.

Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; and Zhao, J. L. 2014. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In *WAIM'14*.