SMATE: Semi-Supervised Spatio-Temporal Representation Learning on Multivariate Time Series
Jingwei Zuo, Karine Zeitouni, Yehia Taher

To cite this version:
Jingwei Zuo, Karine Zeitouni, Yehia Taher. SMATE: Semi-Supervised Spatio-Temporal Representation Learning on Multivariate Time Series. 2021 IEEE International Conference on Data Mining (ICDM), Dec 2021, Auckland, Australia. pp.1565-1570, 10.1109/ICDM51629.2021.00206. hal-04371666

HAL Id: hal-04371666
https://hal.science/hal-04371666v1
Submitted on 3 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License
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) has attracted widespread attention in recent years. In particular, label shortage is a real challenge for the classification task on MTS, considering its complex dimensional and sequential data structure. Unlike self-training and positive unlabeled learning that rely on distance-based classifiers, in this paper, we propose SMATE, a novel semi-supervised model for learning the interpretable Spatio-Temporal representation from weakly labeled MTS. We validate empirically the learned representation on 30 public datasets from the UEA MTS archive. We compare it with 13 state-of-the-art baseline methods for fully supervised tasks and four baselines for semi-supervised tasks. The results show the reliability and efficiency of our proposed method.

Index Terms—Neural Networks, Multivariate Time Series, Semi-supervised Learning, Representation Learning

I. INTRODUCTION

Most Multivariate Time Series (MTS) data, such as sensor readings, are labeled during the data collection process. The post-labeling of MTS is much more costly than classic data (e.g., images, 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 to fully supervised algorithms. The previous studies on weak-label Time Series (TS) learning are usually based on self-training [1] or Positive Unlabeled Learning [2] with a carefully designed distance measure [3] or stopping criterion [4] to learn the pseudo-labels. Besides, they mostly focus on the Univariate Time Series (UTS) with the One-Nearest-Neighbor (1NN) classifier on raw data space, which is widely outpaced by today’s advanced approaches [5], such as Deep Neural Networks (DNNs) [6] or ensemble methods [5].

From Univariate Time Series (UTS) to Multivariate Time Series (MTS), traditional methods usually combine the compact and effective features from different variables, such as Shapelet Ensemble [7], [8], global discriminative patterns [9], or bag-of-patterns [10], [11]. However, the predefined features usually fail to capture MTS essentials: the temporal dependency and the interactions of multiple variables (i.e., spatial interactions). Recently, some deep learning-based methods were proposed to capture the MTS features with various network structures [12], [13], showing promising performance on MTS classification tasks. However, the above-mentioned methods are mostly fully supervised, and rarely consider the label shortage issue when building the MTS classifier.

The recent research turns to Representation Learning [14] when handling weakly labeled MTS, which allows learning low dimensional embeddings in an unsupervised manner, such as using triplet loss [15] to form the embedding space, then even an SVM classifier is powerful enough on the learned representation [15]. However, existing techniques suffer from three major issues. First, the interactions between the MTS variables are generally computed on the entire 1-D series, ignoring the fact that the local spatial interactions at the sub-sequence level may evolve in the dynamic sequence, that is spatial dynamics. Second, the representation learned in a pure unsupervised manner depends mostly on the loss function selection. As no label information is utilized to learn the representation [15], there is a risk that it deviates from the true features, thus affecting the classifier performance. Third, they rather employ deep learning as a blind box and do not focus on the interpretability of the learned representation.

Therefore, to handle both the MTS complex structure and the label shortage problem, we propose SMATE, Semi-supervised Spatio-temporal representation learning on MultivAriate Time SEries. An auto-encoder based structure allows mapping the MTS samples from raw features space $\mathcal{X}$ to low dimensional embedding space $\mathcal{H}$. A Spatial Modeling Block combined with a multi-layer convolutional network captures the spatial dynamics, whereas a GRU-based structure extracts the temporal dynamic features. Thereby, SMATE is capable of compressing the essential Spatio-temporal characteristics of MTS samples into low-dimensional embeddings. On top of this embedding space $\mathcal{H}$, we propose a semi-supervised three-step regularization process to bring the model to learn class-separable representations, where both the labeled and unlabeled samples contribute to the model’s optimization. This regularization process comes with the capability of visualization at each step, making SMATE interpretable.

We summarize the paper’s main contributions as follows:

- **Spatio-temporal dynamic features in MTS**: We claim and demonstrate that the temporal dependency and the evolution of the spatial interactions (spatial dynamics) are important for building a reliable MTS embedding.
- **Weak supervision of representation learning**: With limited labeled data, SMATE can learn reliable class-separable MTS representations for downstream tasks, such as MTS classification (MTSC).
- **Interpretable MTS embedding learning**: SMATE allows for visual interpretability, not only from the class-
separable representations but also in each step of the semi-supervised regularization process.

- **Extensive experiments on the MTS datasets:** The experiments are carried out on 30 MTS datasets from different application domains. The detailed evaluation with 13 supervised baselines and four semi-supervised work is provided, which shows the effectiveness and the efficiency of SMATE over the state of the art.

The rest of this paper starts with a review of the most related work. Then, we formulate the problems of the paper. Later, we present in detail our proposal SMATE, which is followed by the experiments on real-life datasets and the conclusion.

II. RELATED WORK

**A. Multivariate Time Series Representation Learning**

**Definition 1 (multivariate time series):** A multivariate time series (MTS) \( x \in \mathbb{R}^{T \times M} \) is a sequence of real-valued vectors: \( x=(x_1, x_2, ..., x_t, ..., x_T) \), where \( x_t \in \mathbb{R}^M \), \( M \) is the variable number. When \( M = 1 \), we call it *univariate time series* (UTS).

A natural way to learn MTS representation is to extend the approaches developed earlier on UTS [16–18]. For instance, [7] combines Shapelet representation from different variables to build an ensemble-like learner. Similarly, SMTS [10] and WEASEL+MUSE [11] adopt the symbolic features with the Bag-of-Patterns concept to model the variable’s relationship.

Different from the interpretable feature-based representations, various deep learning models are proposed to capture the spatial interactions in MTS. MC-DCNN [15] extracts firstly 1D-CNN features from each variable, then combines them with a Fully Connected (FC) Layer. Whereas the authors in [19] abandon the combination strategy but apply directly 1D-CNN features with the cross-attention mechanism in CA-SFCN [6] enhanced the dependencies captured by 1D-CNN on both temporal and spatial axes. Last but not least, the hybrid LSTM-CNN structure is capable of extracting both local and long-term features.

Various work, such as the Squeeze-and-Excitation block in MLSTM-FCN [12] or the multi-view learning-like module in TapNet [20], enhanced the hybrid structure via modeling the spatial interactions. However, the interactions are generally computed at the sequence level, ignoring the fact that the local spatial interactions at the sub-sequence level may evolve in the dynamic sequence, i.e., spatial dynamics. Moreover, they are all fully supervised, requiring huge labels during the training process. Also, the learned representations are result-oriented (e.g., pursuing higher accuracy), with less focus on the interpretability, considered by the data mining community.

**B. Semi-supervised Learning on Time Series**

The pioneering work [1], [3] on Semi-supervised TS Learning are based on self-training or Positive Unlabeled Learning [2] with the Nearest-Neighbor (1NN) classifier and a carefully designed distance, such as DTW [1] or DTW-D [3], and optimized stopping criterion [4] for importing the pseudo-labels. Those work are followed by [21] for wider contexts.

Though not mentioned in their papers, the self-training framework is extensible from the UTS to the MTS setting by using an adapted distance [22], such as DTW_D or DTW_A. However, under more complex scenarios nowadays, such as 30 UEA MTS datasets [23] collected from different domains, the distance-based classifiers show limited performance [5].

Learning MTS representations [14] in a weakly supervised setting draws much attention recently. Unsupervised Scalable Representation Learning (USRL) described in [15] combines causal dilated convolutions with triplet loss for contrastive learning. Similarly, authors in [21] adopt Multi-Task Learning (MTL) to learn the self-supervised UTS features from an auxiliary forecasting task. The recent work Semi-TapNet [20] proposes an Attentional Prototype Network to learn from the unlabeled samples. However, the above-mentioned approaches do not explore thoroughly both the labeled samples and the Spatio-temporal dynamics in MTS. We show in Table I the comparison of the methods for learning MTS representation.

**TABLE I: Existing methods for learning MTS Representation**

| Method       | SMTS | MUSE | Shapedet | USRL | TapNet | MLSTM-FCN | CA-SFCN | SICN | IINW | DTW | SMATE |
|--------------|------|------|----------|------|--------|-----------|---------|------|------|-----|-------|
| Temporal Dynamics | ✓    | ✓    | ✓        | ✓    | ✓      | -         | ✓       | -    | -    | ✓   | -     |
| Spatial Dynamics | ✓    | ✓    | ✓        | ✓    | ✓      | -         | ✓       | ✓    | -    | ✓   | -     |
| Interpretable Representation | ✓    | ✓    | ✓        | ✓    | ✓      | -         | ✓       | ✓    | -    | ✓   | -     |
| Label Shortage | -    | -    |          | -    | -      | -         |         |      |      | ✓   | -     |

III. PROBLEM FORMULATION

In this section, we formulate the Spatio-temporal dynamics learning and semi-supervised classification problems for MTS. Table [II] summarizes the notations used in the paper.

**TABLE II: Notation**

| Notation | Description |
|----------|-------------|
| M, M_T, M_u | dataset, labeled portion, unlabeled portion |
| T, M, N | MTS length, variable numbers, dataset size |
| L, D | embedding length, embedding dimension size |
| m, D | temporal window size, embedding pool size |
| x, h | MTS instance, latent embedding |
| \( \theta \) | variable/spatial interaction |
| \( \vartheta \) | general parameters to be optimized |

**A. Spatio-temporal Representation for MTS**

The Spatio-temporal modeling of MTS requires considering both the temporal dependency \( p(x_t|x_{t-1}) \) and the spatial interactions between the variables. Previous studies [12], [20] usually consider the spatial interactions at the sequence level: \( s = \{ x^t \times x^t \} \in \mathbb{R}^M \), where \( x^t, x^t \in \mathbb{R}^{T \times 1} \), \( x \) indicates the interactions between the variables. However, the local spatial interactions at the sub-sequence level may evolve in the dynamic sequence, where \( m \) is the window size. To illustrate, given the system status at time \( t \) in MTS, it is not only decided by the local value \( x_t \in \mathbb{R}^M \) given a temporal status, but also by its neighbor values \( x_{t-m/2} : x_{t+m/2} \in \mathbb{R}^M \times m \), which brings a spatial correlation matrix on temporal neighbors and spatial variables given a spatial status \( s_t \in \mathbb{R}^M \).

Therefore, given a sample \( x \in \mathbb{R}^{T \times M} \) in raw space \( \mathcal{X} \), the Spatio-temporal representation learning for MTS is to learn a low-dimensional representation \( h \in \mathbb{R}^{L \times D} \) on embedding.
space $\mathcal{H}$, integrating both temporal dynamic $p(x_t'|x_t)$ and spatial dynamic $p(x_t'|s_t)$. The item dynamic refers to the unstable system status within the evolving multivariate sequential data.

**B. Semi-Supervised Learning on MTS**

**Definition 2 (weak-label MTS dataset):** A weakly labeled MTS dataset $D = \{D_l, D_u\}$ contains both labeled and unlabeled MTS samples:

$$D_l = \{x_i, y_i\}_{i=1}^{N_l}, \quad D_u = \{x_i\}_{i=1}^{N_u(1-r)}$$

$r \ (0 \leq r \leq 1)$ indicates the ratio of the labeled samples in $D$ of size $N$, $y_i$ is the annotation of the labeled instance $x_i$.

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 $D_l$ and further unsupervised adjustment/optimization from $D_u$.

**IV. PROPOSAL: SMATE**

In this section, we introduce SMATE, which captures the essential characteristics of MTS and allows learning an interpretable representation space in a semi-supervised manner.

**A. Global Structure of SMATE**

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

As shown in Figure [1], a two-channel encoder explores both spatial and temporal dynamics and embeds the input MTS into a low-dimensional representation space, where the embedded samples are sparsely distributed with the reconstruction-based optimization. On the unsupervised embedding space, a three-step regularization is applied to learn class-separable embeddings. The class centroids are regularized by labeled and unlabeled samples, showing interpretability over the representation space. Finally, the model is jointly optimized by the reconstruction objective and the regularization objective.

**B. Spatial Modeling Block (SMB)**

Firstly, we introduce a novel module, Spatial Modeling Block (SMB), to capture the spatial interactions at subsequences levels. As shown in Figure [2] SMB takes as input an MTS representation $h = \{h_i\}_{i=1}^{T} \in \mathcal{R}^{T \times d} \ (d = M$ for the first block in spatial encoding channel), followed by a one-dimensional average pooling layer on each variable $h_j \in \mathcal{R}^{T \times 1}$, encoding the temporal neighbors into the horizontal system status $s_h(i) = \text{avg}(\{h_{i-m/2} : h_{i+m/2}\})$, where $i$ is the time tick, $m$ is the window size. Then the Fully Connected (FC) layers allow first the interaction of the horizontal system status $s_h$ in the vertical direction via a low-dimensional 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 $s = \{s_i\}_{i=1}^{T}$, where $s_i \in \mathcal{R}^d$, representing the interaction weights for the vector $h_i \in \mathcal{R}^d$. The output of SMB is described by $h' = h \odot s$, with the calibrated weights for each 1-D TS segment, where $\odot$ is the element-wise multiplication.

**C. Spatio-Temporal Encoding on MTS**

Given $x \in \mathcal{R}^{T \times M}$, the low-dimensional representation $h \in \mathcal{R}^{L \times D}$ embeds the Spatio-temporal features of $x$ by a neural network-based function $f_h(x)$. The low-dimensional embedding brings dramatic improvement on both the efficiency and accuracy for classification tasks [20], as the classifier is not impaired by the redundant information in raw data.

As shown in Figure [1] for the temporal channel, among different variants of the recurrent neural networks (RNN), we specifically consider Gated Recurrent Units (GRUs) that mitigate the vanishing gradient problem. An update gate $z_t$ and a reset gate $r_t$ control the hidden state $h_t \in \mathcal{R}^d$ with the observation $x_t \in \mathcal{R}^M$ and the previous hidden state $h_{t-1} \in \mathcal{R}^d$, where $d_g$ is the output dimension of GRUs. The update functions are as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (1)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2)$$

$$h_t = (1-z_t) \odot h_{t-1} + z_t \odot \text{tanh}(W_x x_t + U_x h_{t-1} \odot r_t) + b_h \quad (3)$$

where $W_x, U_x, b_x \ (x \in \{r, z, h\})$ are model parameters, $\sigma$ is the sigmoid function, $\odot$ is the element-wise multiplication. Three GRUs are cascaded with a 1-D pooling layer to output $h(T) \in \mathcal{R}^{L \times d_p}$, where $L=\mathcal{T}/\mathcal{P}$, $\mathcal{P}$ is the pool/sampling size.

For the spatial channel, we define the convolutional module:

$$h'(l) = \text{SM}(h(l)) \quad (4)$$

$$h(l+1) = \text{ReLU}(BN(W \odot h'(l) + b)) \quad (5)$$

where $l \ (0 \leq l < 3)$ is the module number, $h(0)=x \in \mathcal{R}^{T \times M}$, $h(l) \in \mathcal{R}^{T \times d_c}$, $d_c$ is the filter number, $W$ is the 1-D convolutional kernel of size $m$, $\odot$ is the convolution operator. Within each of the three modules, the SMB firstly calibrates the interaction weights for each 1-D segment and outputs $h' \in \mathcal{R}^{T \times d}$. Then a 1-D convolutional layer concatenated with Batch Normalization and Rectified Linear Units (ReLU) is deployed. Similar to the temporal channel, a 1-D pooling layer is applied after the last convolutional block to output $h(S) \in \mathcal{R}^{L \times d_s}$. Finally, we output the combined spatial and temporal features $h_{concat} = \text{concat}(h(T), h(S)) \in \mathcal{R}^{L \times (d_s + d_t)}$ and apply two FC layers to get the Spatio-temporal embedding $h \in \mathcal{R}^{L \times D}$. The matrix representation captures the Spatio-temporal features and facilitates the MTS restoration.

**D. Joint Model Optimization**

As shown in Figure [1] since the representation learned via an autoencoder-based structure generally has a sparse distribution of class-specific samples [14], the unsupervised training derived from the reconstruction objective does not consider thoroughly the inner relation between class-specific samples but focus on the restoration performance from the embeddings. To address the issue, we propose a joint model optimization that integrates the temporal reconstruction and the three-step regularization objectives. Specifically, the joint optimization combines both the labeled and unlabeled samples to learn the class-specific clusters on the embedding space.

Firstly, we define the temporal reconstruction loss as:

$$L_R = \sum_t \| x_t - \tilde{x}_t \|_2 \quad (6)$$
Fig. 1: Model Structure of SMATE

where \(x_t, \hat{x}_t \in \mathcal{R}^M\), corresponding to the observations in the raw and reconstructed MTS instances \(x\) and \(\hat{x}\).

Then, the three-step regularization approaches the embeddings within the class-specific clusters to the virtual class centroids, which are trained progressively.

**Step 1 Supervised Centroids Initialization:** The class centroids are initialized by the class-specific embeddings. Given the labeled training set \(\mathcal{D}_l = \{X^k\}_{k=1}^K\) where \(K\) is the class number, \(X^k \in \mathcal{R}^{N_k \times T \times M}\) is a sample collection of class \(k\), \(N_k\) is the sample number of class \(k\). Then the embedding set 
\(H^k = f_{\theta}(X^k) \in \mathcal{R}^{N_k \times L \times D}\) initializes the class centroid \(c_k\) by:
\[
c_k = mean(H^k), \quad c_k \in \mathcal{R}^{L \times D}
\]  
(7)

**Step 2 Supervised Centroids Adjustment:** Once the centroids are initialized, we can make the supervised adjustment since the distance-based class probability allows to make the contribution of individual samples on the centroid’s decision. In other words, the centroid \(c_k\) is affected by larger contribution weights brought by nearby samples of class \(k\). Let \(x_k^i \in \mathcal{R}^{T \times M}\) be a time series of class \(k\), we define the weight of \(x_k^i\) to \(c_k\) as the inverse Euclidean Distance (ED) between the embedding \(h_k^i = f_{\theta}(x_k^i) \in \mathcal{R}^{L \times D}\) and the centroid \(c_k\):
\[
W_{k,i} = 1 - \frac{ED(h_k^i, c_k)}{\sum_{j=1}^{K} ED(h_k^j, c_j)}
\]  
(8)

Then the class centroid \(c_k\) can be adjusted accordingly by the labeled samples within the class-specific cluster:
\[
c_k = \sum_{i=1}^{N_k} W_{k,i} \cdot h_k^i, \quad h_k^i \in H^k
\]  
(9)

**Step 3 Unsupervised Centroids Adjustment:** Apart from the reconstruction-based optimisation, the unlabeled sample \(\hat{x}_i\) in \(\mathcal{D}_u = \{ \hat{x}_i \}_{i=1}^{N_u} \) is capable of adjusting the centroid \(c_k\) via the propagated label from the distance-based class probability:
\[
\hat{p}_0(y = k|\hat{x}_i) = 1 - \frac{ED(f_{\theta}(\hat{x}_i), c_k)}{\sum_{j=1}^{K} ED(f_{\theta}(\hat{x}_i), c_j)}
\]  
(10)

The unlabeled sample \(\hat{x}_i\) will be then integrated into the class-specific cluster with the highest probability. We can further adjust the class centroid \(c_k\) considering the unlabeled samples:
\[
c_k = \frac{N_k}{N_k + \hat{N}_k} \sum_{i=1}^{N_k} W_{k,i} \cdot h_k^i + \frac{\hat{N}_k}{N_k + \hat{N}_k} \sum_{i=1}^{\hat{N}_k} \hat{p}_{k,i} \cdot \hat{h}_k^i
\]  
(11)

where \(\hat{h}_k^i = f_{\theta}(\hat{x}_k^i), \hat{N}_k\) is the number of samples of class \(k\) in \(\mathcal{D}_u\) with the propagated label.

The class centroids are initialized and adjusted by both labeled and unlabeled samples, from which we formalize the regularization loss derived from the labeled samples:
\[
L_{Reg}(\theta) = - \sum_k logW_{\theta}(y = k|x)
\]  
(12)

As both the reconstruction and regularization losses are normalized, we define the global optimization objective as:
\[
\min_{\theta} (L_R + \lambda L_{Reg})
\]  
(13)

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-separable embeddings. In practice, SMATE is not sensitive to \(\lambda\); then for all the experiments, we set \(\lambda = 1\).

V. Experiments

A. Experimental setup

The model was trained using the Adam optimizer on a single Tesla V100 GPU of 32 Go memory with CUDA 10.2. We train an SVM classifier with radial basis function kernel on the learned embedding space, which is evaluated on the newly released UEA archive \([23]\) for supervised analysis, where the datasets \{ArticularyWordR, Epilepsy, Heartbeat, SelfRegulationSCP1\} from four different domains are adopted for semi-supervised study.

B. Classification Performance Evaluation

We use the accuracy as the default metric \([5]\) for the supervised tasks. We also report the number of Win/Ties and the average rank \([20]\) of different methods.

1) Comparison Methods:

- Distance-based Nearest Neighbor (1NN) on non-normalized (non-norm) or normalized (norm) MTS \([22]\): 1NN-ED; 1NN-DTW\(_T\); 1NN-DTW\(_D\); 1NN-DTW\(_A\).
- Bag-of-patterns classifier. WEASEL+MUSE \([11]\).
- Deep Learning-based classifier. USRL \([15]\); TapNet \([20]\); MLSTM-FCN \([12]\); CA-SFCN \([6]\); SMATE\(_NR\): SMATE without supervised Regularization, instead, a Softmax layer is applied on the embedding.

\(^2\)The source code: https://github.com/JingweiZuo/SMATE
\(^3\)The datasets can be found in www.timeseriesclassification.com
2) Results Analysis: Table III shows the results comparison with the baselines. “N/A” indicates the model is not applicable due to memory overflow. Overall, SMATE defends its reliability with 11 Wins/Ties and the highest average rank of 3.85 among all the baselines. The current state-of-the-art deep learning methods (TapNet, CA-SFCN) and the powerful data mining method (WEASEL+MUSE) have close ranks (4.73/5.45/4.66). Besides, USRL and SMATE-SF perform much worse than SMATE with the same SVM classifier, confirming the reliability of our supervised regularization on the embedding space. Moreover, SMATE achieves the best performance among the baselines on all the datasets of EEG/MEG applications (FaceDetection, FingerMovements, Hand-MovementDirection, MotorImagery, SelfRegulationSCP1, SelfRegulationSCP2), where the signals (i.e., variables) generally have strong and dynamic dependencies with each other. The spatial dynamic interactions could be essential characteristics that SMATE has successfully captured.

C. Semi-supervised Classification Performance

For semi-supervised tasks, we evaluate the classifier’s accuracy at different supervision levels. For comparison, we applied one classic model 1NN-DTW-D and three recently proposed semi-supervised deep learning models: USRL, Semi-TapNet and MTL. Since 1NN-DTW-D and MTL are initially designed for UTS, we adapt them by:

- Adopting DTW as distance in INN-DTW-D.
- Updating the MTS network optimization metrics in MTL.

The results in Figure 5 shows the superiority of SMATE over the baselines. In a Motion Recognition task (Articulatory-Work), from 10% labeled training set to fully labeled one, the accuracy of SMATE varies only by 0.046, compared to INN-DTW-D (0.264), USRL (0.286), Semi-TapNet (0.151) and MTL (0.225), showing that SMATE is capable of learning a class-separable representation under weak supervision. This conclusion is also demonstrated in EEG/MEG applications (SelfRegulationsCP1), with 10% labeled samples, SMATE obtained a higher accuracy (0.781) than fully supervised INN-DTW-D (0.775), USRL (0.771), Semi-TapNet (0.739) and MTL (0.730). In an Audio Spectra task (Heartbeat), though the fully supervised accuracy of SMATE (0.741) is not as good as Semi-TapNet (0.751), the weakly supervised SMATE with 10% labeled samples performs the best among all the models, indicating the reliability of the semi-supervised model.

D. Visualization & Interpretation of the Representation Space

Apart from the thorough exploration of the weakly labeled samples, the representation space learned via SMATE shows good interpretability compared to the traditional Deep Learning models.

We show in Figure 6 that t-SNE visualization of the representation space for the Epilepsy dataset, which contains four human activities: Walking while gesturing, Walking slowly, Walking fast, Walking normally. The 137 samples in the training set are projected to the representation space with 10% labels adopted for training. We show respectively the space visualization at each regularization step. The results suggest that the representation space is interpretable for not only the effect of the weak supervision at each step but also the classification results. For instance, in Figure 7, three samples of Walking while gesturing stay close to the class centroid of Walking fast, which may lead to the misclassification of certain samples in the two classes. To improve the classifier, more labels of the two classes in the training set can be added.

E. Performance of Spatial Modeling Block (SMB)

To validate the Spatial Modeling Block (SMB), firstly, we compare the classification accuracy of SMATE with or without integrating SMB on the 27 datasets that SMATE has successfully executed. Then we rebuild SMATE by replacing SMB with the following modules in the state-of-the-art work which learn the variable relationships of MTS: Random Dimension Permutation (RDP) in TapNet and Squeeze-and-Excitation (SE) in MLSTM-FCN. Briefly, SMATE-SMB achieves [17 Wins/8 Ties/2 Losses] to SMATE-NonSMB, indicating that SMB contributes to a better MTS representation. Besides, SMB performs better than other modules: [14 Wins/8 Ties/5 Losses] to SE, [12 Wins/9 Ties/6 Losses] to RDP. RDP performs relatively better than SE, as a set of grouped variables produced by RDP provides various MTS views, allowing exploring the interactions between the subsets of all variables more thoroughly. However, extra parameters for variable groups are introduced. SE is a parameter-free module but considers each variable has a unique and stable state when interacting with others, which ignores the dynamic features in time series. 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 variable interactions at the sequence level.

VI. CONCLUSION

In this paper, we proposed SMATE, to learn the Spatial-temporal representation on weakly-labeled multivariate time series. The weak supervision on the embedding space allows building a reliable classifier, which is extremely valuable in real-life scenarios with label shortage issues. The results show that the evolving variable interactions (i.e., spatial dynamics) play an essential role in modeling multivariate time series. Moreover, SMATE allows for visual interpretability in both the learned representation and the semi-supervised representation learning process. Our future work will be oriented towards extending SMATE to support multivariate time series with missing values and unequal length in more realistic scenarios.

ACKNOWLEDGEMENTS

This research was supported by DATAIA convergence institute as part of the Programme d’Investissement d’Avenir (ANR-17-CONV-0003) operated by DAVID Lab, UVSQ, Université Paris-Saclay. The authors would like to thank Anthony Bagnall and his team for providing the community with valuable datasets and source codes in the UEA & UCR Time Series Classification Repository. The authors would also like to thank Eamonn Keogh for his careful review and remarks.
TABLE III: Performance Comparison for MTS classification over UEA MTS archive

| Dataset                  | USRL | TopNet | TapNet | 1NN-DTW | MTL | SMATE | Supervised Ratio | Semi-TapNet | SMATE | USRL | 1NN-DTW | MTL | SMATE | Supervised Ratio | Semi-TapNet | SMATE | USRL | 1NN-DTW | MTL | SMATE |
|--------------------------|------|--------|--------|---------|-----|-------|------------------|-------------|-------|------|---------|-----|-------|------------------|-------------|-------|------|---------|-----|-------|
| UWaveGestureLibrary      | 0.979| 0.954  | 0.971  | 0.984   | 0.99| 0.985 | 0.85             | 0.91        | 0.985 | 0.979| 0.934   | 0.98| 0.985 | 0.85             | 0.91        | 0.985 | 0.979| 0.934   | 0.98| 0.985 |
|walking while gesturing   | 1.0  | 1.0    | 1.0    | 1.0     | 1   | 1     | 1.0              | 1.0         | 1.0   | 1.0 | 1.0     | 1   | 1     | 1.0              | 1.0         | 1.0   | 1.0 | 1.0     | 1   | 1     |
|walking normal            | 0.969| 0.969  | 0.969  | 0.969   | 0.99| 0.969 | 0.969            | 0.969       | 0.969 | 0.969| 0.969   | 0.99| 0.969 | 0.969            | 0.969       | 0.969 | 0.969| 0.969   | 0.99| 0.969 |
|walking fast              | 0.979| 0.979  | 0.979  | 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 |
|MotorImagery              | 0.916| 0.916  | 0.916  | 0.916   | 0.94| 0.916 | 0.916            | 0.916       | 0.916 | 0.916| 0.916   | 0.94| 0.916 | 0.916            | 0.916       | 0.916 | 0.916| 0.916   | 0.94| 0.916 |
|walking slowly            | 0.979| 0.979  | 0.979  | 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 |
|walking normal            | 0.94  | 0.94   | 0.94   | 0.94    | 0.93| 0.94  | 0.94             | 0.94        | 0.94  | 0.94 | 0.94    | 0.93| 0.94  | 0.94             | 0.94        | 0.94  | 0.94 | 0.94    | 0.93| 0.94  |
|walking slowly            | 0.979| 0.979  | 0.979  | 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 | 0.979            | 0.979       | 0.979 | 0.979| 0.979   | 0.99| 0.979 |

REFERENCES

[1] L. Wei and E. Keogh, “Semi-supervised time series classification,” in KDD, 2006, p. 748–753.
[2] M. N. Nguyen, X. L. Li, and S. K. Ng, “Positive unlabeled learning for time series classification,” in IJCAI, 2011, p. 1421–1426.
[3] Y. Chen, B. Hu, and E. Keogh, “DTW-D: Time Series Classification using Divergence between DTW Distances,” in IJCAI, 2013, p. 383–391.
[4] C. A. Ramakanthavanta and D. Wanchsan, “Stopping Criteria Selection for Efficient Semi-supervised Time Series Classification,” Softw. Pract. Exper, vol. 42, 2012, pp. 189–211.
[5] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, “The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” Data Mining and Knowledge Discovery, vol. 35, no. 2, pp. 401–449, 2021.
[6] Y. Hao and H. Cao, “A New Attention Mechanism to Classify Multi-Label Time Series,” in IJCAI, 2019, pp. 307–315.
[7] M. S. Cetin, A. Mueen, and V. D. Calhoun, “Shapelet ensemble for large time series with se4tec,” in ICDM, 2014, pp. 298–305.
[8] R. Mousheimish, Y. Taher, and K. Zeitouni, “Automated Learning of Predictive CEP Rules: Bridging the Gap between Data Mining and Complex Event Processing,” in DEBS, 2017, p. 158–169.
[9] A. Ortle, F. Li, W. Song, and S. Li, “Learning Discriminative Virtual Sequences for Time Series Classification,” in CIKM, 2020, p. 2001–2004.
[10] G. Gokcay Baydogan, G. Runger, M. G. Baydogan, and G. Runger, “Learning the great time series classification for multivariate time series classification,” Data Min Knowl Discov, vol. 29, pp. 400–422, 2015.
[11] P. Schäfer and U. Leser, “Semi-supervised Time Series Classification with WEASEL+MUSE,” in AALTD, 2018.
[12] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate Itemsets for time series classification,” Neural Networks, vol. 116, 2019.
[13] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in WAIM, 2014, pp. 298–310.
[14] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE TNNLS, pp. 1798–1828, 2015.
[15] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, “Unsupervised Scalable Representation Learning for Multivariate Time Series,” in NewIPS’19.
[16] J. Zuo, K. Zeitouni, and Y. Taher, “Exploring interpretable features for large time series with se4tec,” in EDBT, 2019, pp. 606–609.
[17] J. Zuo, K. Zeitouni, and Y. Taher, “Incremental and Adaptive Feature Exploration over Time Series Stream,” IEEE BigData’19, pp. 593–602.
[18] J. Zuo, K. Zeitouni, and Y. Taher, “ISETS: Incremental Shapelet Extraction from Time Series Stream,” ECML-PKDD, pp. 790–793, 2019.
[19] J. Yang, M. S. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in IJCAI, 2015, pp. 3995–4001.
[20] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, “TapNet: Multivariate Time Series Classification with Transformer,” in AAI20.
[21] S. Jawed, J. Grabova, and L. Schmidt-Thieme, “Self-Supervised Learning for Semi-Supervised Time Series Classification,” in PAKDD’20.
[22] M. Shokoo-Itka, B. Hu, H. Jin, J. Wang, and E. Keogh, “Generalizing DTW to the multi-dimensional case requires an adaptive approach HHS Public Access,” Data Min Knowl Discov, vol. 31, no. 1, pp. 1–31, 2017.
[23] A. Bagnall, H. A. Dau, J. Linde, J. Large, A. Bostrom, P. Southam, and E. Keogh, “The ua multivariate time series classification archive, 2018,” arXiv preprint arXiv:1811.00075, 2018.

The t-SNE visualization of the representation space for the UWaveGestureLibrary dataset. We set the supervised ratio to 0.1. The colors of the embeddings represent their inherent labels, which are not fully used for training. The class centroids are marked by *.