# DECADE: A Deep Metric Learning Model for Multivariate Time Series

Zhengping Che
University of Southern California
Department of Computer Science
Los Angeles, California 90089
zche@usc.edu

Xinran He
University of Southern California
Department of Computer Science
Los Angeles, California 90089
xinranhe@usc.edu

Ke Xu
University of Southern California
Department of Computer Science
Los Angeles, California 90089
xuk@usc.edu

Yan Liu
University of Southern California
Department of Computer Science
Los Angeles, California 90089
yanliu.cs@usc.edu

## ABSTRACT

Determining similarities (or distance) between *multivariate* time series sequences is a fundamental problem in time series analysis. The complex temporal dependencies and variable lengths of time series make it an extremely challenging task. Most existing work either rely on heuristics which lacks flexibility and theoretical justifications, or build complex algorithms that are not scalable to big data. In this paper, we propose a novel and effective metric learning model for multivariate time series, referred to as *Deep ExpeCted Alignment DistancE* (**DECADE**). It yields a valid distance metric for time series with unequal lengths by sampling from an innovative alignment mechanism, namely *expected alignment*, and captures complex temporal multivariate dependencies in local representation learned by deep networks. On the whole, DECADE can provide valid *data-dependent* distance metric efficiently via end-to-end gradient training. Extensive experiments on both synthetic and application datasets with multivariate time series demonstrate the superior performance of DECADE compared to the state-of-the-art approaches.

## KEYWORDS

Multivariate Time Series, Metric Learning, Deep Learning

**ACM Reference format:**
Zhengping Che, Xinran He, Ke Xu, and Yan Liu. 2017. DECADE: A Deep Metric Learning Model for Multivariate Time Series. In *Proceedings of 3rd SIGKDD Workshop on Mining and Learning from Time Series, Halifax, Nova Scotia, Canada, Aug 14, 2017 (MiLeTS17),* 9 pages.
https://doi.org/10.475/123_4

## 1 INTRODUCTION

Multivariate time series data is ubiquitous in many practical applications, such as health care [25], neuroscience [20], and speech recognition [19]. One of the fundamental problems in time series analysis is measuring the distance (or similarities) between time series sequences. For example, in health care applications, doctors are interested in answering "patients like me", i.e., identifying similar patients in the database as the query patient by comparing their time series records of vital signs and lab measurements. Efficient and accurate similarity and distance measure also serves as the basis for many important tasks including but not limited to search, classification, and clustering [7, 17, 28].

A variety of similarity measures have been developed for time series data, most of which consist of two components [22, 24, 26]: an alignment between two time series which matches the time steps and a predefined *local* distance between feature vector pair at each single time step. The resulted (global) similarity measure is then the sum of all the local distances between the aligned feature pairs. However, existing alignment algorithms are either heuristics without theoretical justifications or too complex to be calculated efficiently. For example, dynamic time warping (DTW) [1], one of the most popular alignment algorithms, does not produce a valid distance metric since it violates the triangle inequality. Global alignment kernel (GAK) [4] produces a positive definite kernel (and thus can produce valid distance metric) only if certain property is satisfied in the local kernel and is computationally expensive on big datasets. Moreover, a set of distinctive properties of *multivariate* time series data, such as complex temporal dependencies, high dimensionality, large scale, and irregular sampling [12], makes standard data-independent local distances, such as Euclidean distance, insufficient to measure similarities. In brief, there is no *universal* similarity metric that can work best across all time series applications [13], and learning a powerful *data-dependent* distance metric is an essential step to the success of many learning tasks on different multivariate time series data.

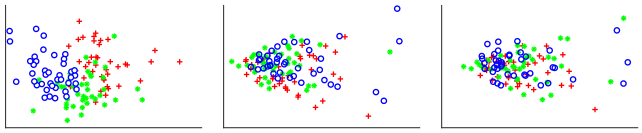
To demonstrate the necessity of learning data-dependent distance and motivate our proposed model, we construct a synthetic dataset with 150 multivariate time series of three classes generated from multivariate non-linear base functions with random temporal shifts. We compared the similarity computed by MDTW and GAK with $L_2$ distance to our proposed method by visualizing the 2-dimension embedding using multidimensional scaling (MDS) [2]. Without learning any metric, the samples from different classes are


*MiLeTS17, Aug 14, 2017, Halifax, Nova Scotia, Canada*

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

mixed together in Figure 1(b) and 1(c). On the contrary, as shown in Figure 1(a), our proposed data-dependent metric makes the three time series clusters quite distinguishable even in low dimensional space.

In this paper, we develop a novel multivariate time series metric learning framework called *Deep ExpeCted Alignment DistancE* (DECADE). we utilize deep networks to capture the complex temporal dependencies in data-dependent local distance for multivariate time series. As shown by recent development in deep learning on metric learning for other types of structured data [6, 9, 11, 29], deep network models have much larger model capacity than other methods to learn nonlinear metric. To make training procedure efficiency and ensure the obtained metric is a valid (global) distance metric, we propose a new alignment method namely *expected alignment*. Instead of taking one single best warping path in DTW and all possible alignments in GAK, the expected alignment averages the distance between aligned time series over all warping paths of proper length. We prove that DECADE yields a valid metric satisfying triangular inequality while only requiring that the local distance is a valid metric. Moreover, the expected alignment can be efficiently computed by sampling a few warping paths. Lastly, DECADE is flexible enough that any existing metric learning framework, such as the large margin approach, can be plugged in so that complex data-dependent distance metric can be effectively learned via end-to-end gradient training. We compare DECADE to several state-of-the-art time series metrics and similarities on both synthetic and real-world health care datasets. Experiment results on tasks including visualization and classification demonstrate the superiority of our approach.



(a) Proposed Metric (b) *MDTW* (c) *GAK*

**Figure 1: Visualizations of time series embedding in 2 dimensions via multidimensional scaling. Different colors and markers refer to different classes.**

## 2 RELATED WORK

In this section, we review several representative time series metric learning models and compare them to DECADE in terms of three aspects: which *local* distance is used; how the time series are *aligned*; and whether the global similarity measure is a *valid* distance. A summmary is shown in Table 1.

Some commonly used methods take predefined local distance, such as multivariate dynamic time warping (MDTW) [1], global alignment kernel (GAK) [4], and multiple sequence alignment (MSA) [10]. A few other works aim to learn a linear local distance metric. For example, ML-TSA [14] learns a linear Mahalanobis local distance assuming the ground truth alignment is given. LDMLT-TS [16] takes LogDet divergence to learn a linear local metric under the best alignment path. However, standard data-invariant or linear local distance functions may not be sufficient for complex multivariate time series data. Different from these methods, DECADE applies a deep neural network model to learn local distance which enables it to capture high-dimensional correlations and interactions.

**Table 1: Comparisons of common time series similarity measures.**

| | Data-dependent local metric | Considering alignment | Valid metric |
|---|---|---|---|
| *MDTW* | No | Single | No |
| *GAK* | No | Multiple | *Yes*[1] |
| *MSA* | No | Single | Yes |
| *ML-TSA* | Yes | Single | No |
| *LDMLT-TS* | Yes | Single | No |
| *MaLSTM* | Yes (Deep) | No | Yes |
| *DECADE* | Yes (Deep) | Multiple | Yes |

In terms of the alignment method, most existing methods define the global similarity measure under the single best warping path as in MDTW. It not only leads to a invalid metric violating triangular inequality but also results in inefficient training procedure due to the iteration between finding the warping path and optimizing the local distance model. Besides the alignment based approaches, Long Short-Term Memory (LSTM) models are also directly used for modeling time series similarities [18, 21]. The states from last hidden layer are treated as the representations and the $L_1$ or $L_2$ distance on the representations are computed as the global distance. However, these approaches usually focus on the overall patterns and cannot effectively capture the complex temporal dependencies in multivariate time series which can only be exposed by temporal alignment.

For the validity of the global distance, only three approaches besides DECADE produce valid distance metrics. However, GAK and MSA use predefined data-independent local distance thus leading to inaccurate metric due to the incompetence in capturing complex interactions. On the contrary, DECADE produces a valid global distance metric with flexible local distance.

## 3 METHODOLOGY

An ideal global similarity for multivariate time series should have all the following three desired properties: First, the model should have enough capacity to capture complex high-dimensional interactions in multivariate time series accurately. Second, the similarity measure should be a valid distance metric so that it can be used for kernel definition. Third, the proposed similarity measure should be computationally efficient in both training and testing.

In this section, we present our proposed multivariate time series distance metric called *Deep ExpeCted Alignment DistancE* (DECADE) with all three desired properties. We first describe the two major components of DECADE, the expected alignment and the

[1] Constraints on local kernel selection.

deep local distance model, which can capture the complex interactions from time series and ensure the validity of the learned global metric. Then we show how DECADE can be trained efficiently in an end-to-end way using the large margin metric learning framework and back-propagation.

In the rest of this paper, we use bold capital letter, such as $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$, to denote multivariate time series. Here $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{T_X})^{\mathrm{T}} \in \mathbb{R}^{T_X \times P}$ is a multivariate time series of length $T_{\boldsymbol{X}}$ and with $P$ features. Its $t$-th column $\boldsymbol{X}_t \in \mathbb{R}^P$ represents its observation at time step $t$.

### 3.1 Expected alignment

The expected alignment considers the average distance over all possible alignment paths with length between $U_l$ and $U_h$. Here $U_l$ and $U_h$ are two free parameters which can be chosen based on data properties and efficiency requirement and will be discussed later. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two time series of length $T_{\boldsymbol{X}}$ and $T_{\boldsymbol{Y}}$ respectively, and $\mathcal{A}_U$ be the set of all possible alignment paths of length $U$ between $\boldsymbol{X}$ and $\boldsymbol{Y}$, the distance with expected alignment is defined as

$$D_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y}) = \mathbb{E}_{U \in [U_l, U_h]} \left[ \mathbb{E}_{\boldsymbol{A} \in \mathcal{A}_U} \left[ D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})} \right] \right] = \frac{1}{U_h - U_l + 1} \sum_{U=U_l}^{U_h} \frac{1}{|\mathcal{A}_U|} \sum_{\boldsymbol{A} \in \mathcal{A}_U} D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})} \quad (1)$$

where $D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})}$ is the global distance of $\boldsymbol{X}$ and $\boldsymbol{Y}$ on one alignment path $\boldsymbol{A}$.

Before giving the formal definition of $D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})}$, it is necessary to describe how the alignment path is mathematically represented. An *alignment* path $\boldsymbol{A}$ for two time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be represented by a pair of monotonically non-decreasing sequences $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with the same length $U$. This sequence pair corresponds to a mapping between the two time series: For all $t \in \{1, \ldots, U\}$, we have $\alpha_t \in \{1, \ldots, T_{\boldsymbol{X}}\}$ and $\beta_t \in \{1, \ldots, T_{\boldsymbol{Y}}\}$, and we map all the $\boldsymbol{X}_{\alpha_t}$ and $\boldsymbol{Y}_{\beta_t}$. In most cases, some constraints on the alignment are introduced to make a better trade off between the efficiency and accuracy.[2] Given the local distance $d(\cdot, \cdot)$, the distance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ on the alignment $\boldsymbol{A}$ is defined as $D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})} = \sum_{t=1}^{U} d(\boldsymbol{X}_{\alpha_t}, \boldsymbol{Y}_{\beta_t})$. In practice, $d(\cdot, \cdot)$ can be any valid local distance, such as the commonly used squared Euclidean distance $d(\boldsymbol{X}_{\alpha_t}, \boldsymbol{Y}_{\beta_t}) = \|\boldsymbol{X}_{\alpha_t} - \boldsymbol{Y}_{\beta_t}\|_2^2$. In the proposed DECADE model, we combine the neural network based local distance $d_{dnn}(\boldsymbol{X}_{\alpha_t}, \boldsymbol{Y}_{\beta_t})$ described in Section 3.2 to capture the complex interaction of high dimensional time series. Also, using average distance instead of single best alignment path makes expected alignment a valid metric since the alignments are no longer coupled together with the local distance as in MDTW. Considering average over only alignment paths with certain lengths instead of all possible paths as in GAK makes our expected alignment more flexible and efficient. We no longer have constraint on the local kernel such that any valid local distance leads to a valid global distance metric.

[2] For instance, DTW requires (i) constraint on start and end points, $(\alpha_1, \beta_1) = (1, 1)$ and $(\alpha_U, \beta_U) = (T_{\boldsymbol{X}}, T_{\boldsymbol{Y}})$, and (ii) constraint on local smoothness: $(\alpha_{t+1}, \beta_{t+1}) - (\alpha_t, \beta_t) \in \{(1, 0), (0, 1), (1, 1)\}$ for all $t \in \{1, \ldots, U-1\}$.

The remaining question is how we can efficiently compute the distance between two time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ using the expected alignment. Our solution is a sampling based method. Though the number of alignment path is exponential in the length of the alignment, the empirical mean of i.i.d sampled alignment paths converges quite fast, and only polynomial number of samples will be sufficient to guarantee a small error.

The key insight for the sampling method is that we can represent the alignment path $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in an equivalent way: For one time series $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T)^{\mathrm{T}}$ and a vector $\boldsymbol{a} = (a_1, \ldots, a_T) \in \mathbb{N}^{\mathrm{T}}$, we write $\boldsymbol{X}_{\boldsymbol{a}}$ as

$$\boldsymbol{X}_{\boldsymbol{a}} = (\underbrace{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_1}_{a_1 \text{ times}}, \underbrace{\boldsymbol{X}_2, \cdots, \boldsymbol{X}_2}_{a_2 \text{ times}}, \cdots, \underbrace{\boldsymbol{X}_T, \cdots, \boldsymbol{X}_T}_{a_T \text{ times}}) \in \mathbb{R}^{U \times P}$$

where $U = \sum_{i=1}^{T} a_i = \|\boldsymbol{a}\|_1$. $\boldsymbol{X}_{\boldsymbol{a}}$ can be considered as the warped time series of $\boldsymbol{X}$ given an alignment path of length $U$. We use $\boldsymbol{X}_{\boldsymbol{a}}(t)$ to denote the $t$-th entry of $\boldsymbol{X}_{\boldsymbol{a}}$. Thus, one alignment $\boldsymbol{A}$ with sequences $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of length $U$ can also be represented as two vectors: $\boldsymbol{a} \in \mathbb{N}^{T_X}$ and $\boldsymbol{b} \in \mathbb{N}^{T_Y}$, where $\|\boldsymbol{a}\|_1 = \|\boldsymbol{b}\|_1 = U$. It's also noting that $\boldsymbol{X}_{\boldsymbol{a}}(t) = \boldsymbol{X}_{\alpha_t}$ and $\boldsymbol{Y}_{\boldsymbol{b}}(t) = \boldsymbol{Y}_{\beta_t}$ for $t \in \{1, \cdots, U\}$. Moreover, we denote $\mathcal{A}(T, U) = \{\boldsymbol{a} \in \mathbb{N}^T | \|\boldsymbol{a}\|_1 = U\}$. The distance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ under alignment $\boldsymbol{A}$ then can be written as $D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})} = D_{\boldsymbol{a}, \boldsymbol{b}}^{(\boldsymbol{X}, \boldsymbol{Y})} = \sum_{t=1}^{U} d(\boldsymbol{X}_{\boldsymbol{a}}(t), \boldsymbol{Y}_{\boldsymbol{b}}(t))$.

In order to sample the alignments, we first uniformly sample a length $U \in [U_l, U_h]$, then uniformly sample $\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}}, U)$ and $\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}}, U)$ independently. Sampling $\boldsymbol{a}$ can be achieved by uniformly sampling a non-negative integer solution of equation $\sum_{t=1}^{T_X} x_i = U$, which reduces to uniformly choosing $T_{\boldsymbol{X}} - 1$ items from $T_{\boldsymbol{X}} + U - 1$ items. We use the same way to get the sample $\boldsymbol{b}$. After that, we can get the sampled alignment path and compute distance along the path.

Next, we show that the expected alignment produces a valid distance metric given that the local similarity measure is a valid metric satisfying the triangular inequality in Theorem 3.1. It holds especially for our DECADE with the neural network based local distance. Moreover, we show that the convergence guarantee of sampling based method to compute the distance with expected alignment in Theorem 3.2. We leave the proofs in the supplementary.

Theorem 3.1. *When the local similarity measure $d(\boldsymbol{X}_t, \boldsymbol{Y}_{t'})$ is a valid distance metric, the expected alignment produces a valid metric $D_{EA}(\boldsymbol{X}, \boldsymbol{Y})$. Namely, it satisfies all the three following properties:*

*(a) $D_{EA}(\boldsymbol{X}, \boldsymbol{Y}) \geq 0$ (non-negativity),*
*(b) $D_{EA}(\boldsymbol{X}, \boldsymbol{Y}) = D_{EA}(\boldsymbol{Y}, \boldsymbol{X})$ (symmetry), and*
*(c) $D_{EA}(\boldsymbol{X}, \boldsymbol{Y}) + D_{EA}(\boldsymbol{Y}, \boldsymbol{Z}) \geq D_{EA}(\boldsymbol{X}, \boldsymbol{Z})$ (triangle inequality).*

Proof. The first two statements can be immediately seen from the definition. We only need to show that the expected distance satisfies triangular inequality.

Given that the distance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ under alignment $\boldsymbol{A}$ can be written as

$$D_{\boldsymbol{A}}^{(\boldsymbol{X}, \boldsymbol{Y})} = D_{\boldsymbol{a}, \boldsymbol{b}}^{(\boldsymbol{X}, \boldsymbol{Y})} = \sum_{t=1}^{U} d(\boldsymbol{X}_{\boldsymbol{a}}(t), \boldsymbol{Y}_{\boldsymbol{b}}(t))$$

and the number of different alignments with path length $U$ is $|\mathcal{A}(T_{\boldsymbol{X}}, U)| \cdot |\mathcal{A}(T_{\boldsymbol{Y}}, U)|$, we can represent the distance with expected alignment as follows:

$$D_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{U_h - U_l + 1} \sum_{U=U_l}^{U_h} \frac{1}{|\mathcal{A}_U|} \sum_{\boldsymbol{A} \in \mathcal{A}_U} D_{\boldsymbol{A}}^{(\boldsymbol{X},\boldsymbol{Y})}$$
$$= \frac{1}{U_h - U_l + 1} \sum_{U=U_l}^{U_h} \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}}, U)} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}}, U)} D_{\boldsymbol{a},\boldsymbol{b}}^{(\boldsymbol{X},\boldsymbol{Y})}}{|\mathcal{A}(T_{\boldsymbol{X}}, U)| \cdot |\mathcal{A}(T_{\boldsymbol{Y}}, U)|} \quad (2)$$

With the above representation, it is easy to show that the expected alignment distance satisfies the triangular inequality. Consider any fixed length $U$, given Equation 2, we need to show

$$\frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} D_{\boldsymbol{a},\boldsymbol{b}}^{(\boldsymbol{X},\boldsymbol{Y})}}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})|} + \frac{\sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} D_{\boldsymbol{b},\boldsymbol{c}}^{(\boldsymbol{Y},\boldsymbol{Z})}}{|\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$\geq \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} D_{\boldsymbol{a},\boldsymbol{c}}^{(\boldsymbol{X},\boldsymbol{Z})}}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|} \quad (3)$$

We drop the symbol $U$ in $\mathcal{A}(T, U)$ to simplify the notations. To be more specific, we have

$$\frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} D_{\boldsymbol{a},\boldsymbol{b}}^{(\boldsymbol{X},\boldsymbol{Y})}}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})|} + \frac{\sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} D_{\boldsymbol{b},\boldsymbol{c}}^{(\boldsymbol{Y},\boldsymbol{Z})}}{|\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$= \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} \left( D_{\boldsymbol{a},\boldsymbol{b}}^{(\boldsymbol{X},\boldsymbol{Y})} + D_{\boldsymbol{b},\boldsymbol{c}}^{(\boldsymbol{Y},\boldsymbol{Z})} \right)}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$= \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} \sum_{t=1}^{U} (\Xi)}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$\geq \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} \sum_{t=1}^{U} \sum_{\boldsymbol{b} \in \mathcal{A}(T_{\boldsymbol{Y}})} d(\boldsymbol{X}_{\boldsymbol{a}}(t), \boldsymbol{Z}_{\boldsymbol{c}}(t))}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$= \frac{|\mathcal{A}(T_{\boldsymbol{Y}})| \sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} D_{\boldsymbol{a},\boldsymbol{c}}^{(\boldsymbol{X},\boldsymbol{Z})}}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Y}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$
$$= \frac{\sum_{\boldsymbol{a} \in \mathcal{A}(T_{\boldsymbol{X}})} \sum_{\boldsymbol{c} \in \mathcal{A}(T_{\boldsymbol{Z}})} D_{\boldsymbol{a},\boldsymbol{c}}^{(\boldsymbol{X},\boldsymbol{Z})}}{|\mathcal{A}(T_{\boldsymbol{X}})| \cdot |\mathcal{A}(T_{\boldsymbol{Z}})|}$$

where $\Xi = d(\boldsymbol{X}_{\boldsymbol{a}}(t), \boldsymbol{Y}_{\boldsymbol{b}}(t)) + d(\boldsymbol{Y}_{\boldsymbol{b}}(t), \boldsymbol{Z}_{\boldsymbol{c}}(t))$.

Then taking a summation over the left and right side of Equation 3 over all length $U \in [U_l, U_h]$ concludes the proof. □

Theorem 3.2. *Given any two time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ and the local distance is bounded by 1, if we approximate expected alignments with $O\left(\frac{U_h^2}{\varepsilon^3}\right)$ alignment samples, with high probability we have*

$$\left| D_{EA}(\boldsymbol{X}, \boldsymbol{Y}) - \hat{D}_{EA}(\boldsymbol{X}, \boldsymbol{Y}) \right| \leq \varepsilon$$

Proof. First we know $\mathbb{E}\left[\hat{D}_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y})\right] = D_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y})$. Since all alignment are sampled independently, all the distances $D_{\boldsymbol{a}_m,\boldsymbol{b}_m}^{(\boldsymbol{X},\boldsymbol{Y})}$ calculated on the alignment sample $(\boldsymbol{a}_m, \boldsymbol{b}_m)$ are independent. As the local distance is bounded by 1, we have $0 \leq D_{\boldsymbol{a}_m,\boldsymbol{b}_m}^{(\boldsymbol{X},\boldsymbol{Y})} \leq U_h$.

From Hoeffding's inequality [8] we know for $n$ independent random variables $X_1, \ldots, X_n$ bounded by the interval $[0, 1]$, we have

$$\mathcal{P}\left(\bar{X} - \mathbb{E}\left[\bar{X}\right] \geq t\right) \leq \exp\left(-2nt^2\right)$$

Then applying it on the distances $D_{\boldsymbol{a}_m,\boldsymbol{b}_m}^{(\boldsymbol{X},\boldsymbol{Y})}$ and letting $t = \pm\frac{\varepsilon}{U_h}$ leads to

$$\mathcal{P}\left(\left|D_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y}) - \hat{D}_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y})\right| \geq \varepsilon\right) \leq 2\exp\left(\frac{-2n\varepsilon^2}{U_h^2}\right)$$

Thus, with $n = O\left(\frac{U_h^2}{\varepsilon^3}\right)$ alignment samples we have

$$\mathcal{P}\left(\left|D_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y}) - \hat{D}_{\mathrm{EA}}(\boldsymbol{X}, \boldsymbol{Y})\right| \leq \varepsilon\right) = 1 - O\left(\exp(\frac{1}{\varepsilon})\right)$$

□

The assumption in Theorem 3.2 on bounded local distance can be easily satisfied in DECADE. For example, we can use sigmoid function in the output layer, and take the squared Euclidean distance divided by the representation dimensionality $Q$, or cosine similarity, as local distance.

## 3.2 Local representation learning via deep networks

Learning a powerful data dependent local distance metric is vital in the success of designing global similarity measure. To capture the complex dependencies, we utilize local representation learning via deep neural network to obtain a data-dependent local distance. We define the function $f_{DNN}(\cdot) : \mathbb{R}^P \mapsto \mathbb{R}^Q$ as the transformation function via deep network, which maps $P$ dimensional input features to $Q$ dimensional representations. Given a time series $\boldsymbol{X}$, we apply the same deep network (with shared parameters) on the observations $\boldsymbol{X}_t$ at each time step $t$, and take the output of the network as the learned representation at that time step. We use $\tilde{\boldsymbol{X}}_t = f_{DNN}(\boldsymbol{X}_t)$ to denote the learned representation of $\boldsymbol{X}_t$. To compute the local distance of feature vector pair of two time series $\boldsymbol{X}$ and $\boldsymbol{Y}$ at step $t$ and $t'$ respectively, we first use our neural network to carry out the feature transformation and the learned local distance is defined as the squared Euclidean distance between $\tilde{\boldsymbol{X}}_t$ and $\tilde{\boldsymbol{Y}}_{t'}$, i.e., $d_{dnn}(\boldsymbol{X}_t, \boldsymbol{Y}_{t'}) = \|\tilde{\boldsymbol{X}}_t - \tilde{\boldsymbol{Y}}_{t'}\|_2^2$.

It should be noticed that naively combining the above local distance model with DTW or GAK does not lead to a valid distance metric. DTW violates the triangle inequality as it computes the alignment using a single best warping path. GAK, on the contrary, could produce a valid distance metric. However, it requires that $\frac{\kappa}{1+\kappa}$ is positive definite where $\kappa$ is the local kernel [3]. This condition is most likely to be violated if complex deep neural network is used as local kernel.

## 3.3 Learning DECADE via large margin approach

In this section, we take the large margin metric learning framework [27], one of the most widely used metric learning frameworks, as an example to show how we learn complex data-dependent distance metric in DECADE via end-to-end gradient training. The basic idea of large margin metric learning is to reduce the distance between the data instance $\boldsymbol{X}$ and the instances with the same label,

generally referred to as the *targets* of $\boldsymbol{X}$, and increase the distance between $\boldsymbol{X}$ and the instances with different labels, referred to as the *imposters* of $\boldsymbol{X}$. Intuitively, better metrics learnt from data should pull the target neighbors closer and push the imposter neighbors further away. Suppose that we are given a set of $N$ multivariate time series $\{\boldsymbol{X}^{(i)}\}_{i=1}^{N}$ from $C$ different classes. The large margin approach aims at minimizing the objective function

$$\mathcal{L}(D) = \mathcal{L}^{+}(D) + \mathcal{L}^{-}(D) + R(D) \tag{4}$$

where

$$\mathcal{L}^{+}(D) = \sum_{i=1}^{N} \sum_{j \in \mathcal{S}_i^{+}} D^{(i,j)}$$

$$\mathcal{L}^{-}(D) = \lambda \sum_{i=1}^{N} \sum_{j \in \mathcal{S}_i^{+}} \sum_{k \in \mathcal{S}_i^{-}} \left[\delta + D^{(i,j)} - D^{(i,k)}\right]_{+}$$

Here $R(D)$ represents the regularization on distance metric and deep networks; $D^{(i,j)} = D_{\mathrm{EA}}(\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)})$ is the global expected distance between $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$; $\mathcal{S}_i^{+}$ and $\mathcal{S}_i^{-}$ are the sets of selected target and imposter neighbors of $\boldsymbol{X}^{(i)}$, respectively; Hyperparameters $\lambda$ and $\delta$ control the impact of targets and imposters in training; $[x]_{+} = \max\{x, 0\}$ is the hinge loss function.

The designed expected alignment sampling enables end-to-end gradient training of the neural network parameters via backpropagation using mini-batch stochastic gradient descent. During training, we sample mini-batches of target and imposter neighbors, and further sample alignment paths to compute the global distance between time series. As the distance along the sampled alignment path is an unbiased estimator of the expected distance, we can carry out efficient training using stochastic gradient descent. In DTW based approaches, on this contrary, as the best alignment path depends on the local similarity measure, parameter learning involves iteration between optimizing the parameters and finding the optimal alignment path. The iterative approach not only leads to inefficiency in terms of running time but also makes the model more prone to be trapped in local optima.

## 3.4 Efficiency of DECADE

In terms of implementation, Several factors have effects on the computational efficiency of DECADE during both training and testing. First, we need to choose proper numbers of target and imposter neighbors in training procedure for each time series, i.e., the size of $\mathcal{S}_i^{+}$ and $\mathcal{S}_i^{-}$ in Equation 4. In practice, we found that setting a number much smaller than the number of training samples $N$ is enough to provide good performance. In order to make the training procedure more efficient, $\mathcal{S}_i^{+}$ and $\mathcal{S}_i^{-}$ are built based on the original DTW distance before training, and after every several training epochs, $\mathcal{S}_i^{-}$ is updated based on the learned distance at that time. Second, we need to set a proper range for the alignment path length and number of sampled alignments used in DECADE. The validity of DECADE always holds no matter what the length range is set, but the distribution of the alignments can be quite different. In our experiments, we set the upper bound $U_h = O(T)$ where $T$ is the average length of time series in the dataset. The approximation bound from Theorem 3.2 requires $O(U_h^2)$ alignment samples, however, in practice using $O(T)$ samples can provide satisfying performance and is more efficient. Thus, it takes $O\left(T^2\right)$ to compute the global similarity between two time series, which is the same as MDTW and GAK. Also, several speedup tricks such as Sakoe-Chiba band constraints [23] for DTW can also be applied to the alignment sampling process in DECADE.

# 4 EXPERIMENT

In this section, we carry out extensive experiments on three real-world datasets from health care domain to compare the performance of our proposed DECADE to several state-of-the-art time series metric learning methods in classification and visualization.

## 4.1 Datasets

We describe the three real-world datasets used in our experiments with their data properties as follows.

- **EEG**[3] The EEG dataset contains measurements from 64 electrodes placed on the subjects' scalps when different number of stimuli (0, 1, or 2) are exposed to the subjects. The subjects belong to either alcoholic or control group. We sample 436 time series from the original dataset. The resulted dataset has 64 features (one for each electrode) with fixed length 16. Our task is to classify the subjects to six classes (whether the subject belongs to alcoholic or control group and the number of stimuli the subject is exposed to) from the recording of the electrodes.
- **PhysioNet**[4] The PhysioNet dataset has 918 irregularly sampled time series with 34 features from the first 48 hours (i.e., with a fixed temporal length of 48) of intensive care unit (ICU) stays after downsampling. We conduct the mortality prediction task, which is a binary classification task to predict in-hospital mortality of the patients.
- **ICU** The ICU dataset, which is firstly introduced in [12], consists of physiologic measurements recorded by clinical staff during the delivery of care in intensive care units at a major urban hospital for one week. We take a subset of this dataset which includes 1734 time series with 13 features. The length of time series varies from 24 to 36. Similar to the PhysioNet dataset, we conduct the mortality prediction task on this dataset.

In both PhysioNet and ICU datasets, the ratio of positive samples is 50%. It should be noticed that these two datasets have lots of missing values and irregular samples and we applied the commonly used last observation carried forward imputation method [5]. The noisy observations make the prediction tasks very challenging where simple similarity measures may not perform well. There are no missing values in EEG dataset but the number of features is larger than the length of time series, which also affects the performance of simple similarity measures.

## 4.2 Algorithms

*Baseline methods.* We test several multivariate time series similarity measures as baselines in our experiments. We group the

[3] https://archive.ics.uci.edu/ml/datasets/EEG+Database
[4] http://physionet.org/challenge/2012/

baselines based on whether local distances are learned or not as follows.

The first three methods use predefined data-independent local distances.

- *MDTW*: Multivariate dynamic time warping.
- *GAK*: Global alignment kernel [4]. We use the suggested settings of the hyperparameters from the original paper and $D_{\text{GAK}}^{(i,j)} = \frac{K_{\text{GAK}}(i,j)}{\sqrt{K_{\text{GAK}}(i,i)K_{\text{GAK}}(j,j)}}$ as the global distance from GAK kernel.
- *MSA*: Multiple sequence alignment in [10] with $L_2$ distance as the local distance model.

The following five baselines learn data-dependent similarity measures. Notice that all methods in this group use label information in training.

- *ML-TSA*: Metric Learning for Temporal Sequence Alignment proposed by [14] with iterations between learning the local metric and finding the optimal alignments as no ground truth alignment is provided in our dataset.
- *LDMLT-TS*: Method from [16] with default hyperparameter settings in their implementation.
- *MaLSTM*: Method proposed in [18].
- *MSA-NN*: Extension of *MSA* which iterates between finding the best alignment from *MSA* and optimizing a 2-layer feed forward neural network as local distance.
- *MDTW-NN*: Extension of *MDTW* combined with our learnable deep local distance model proposed in Section 3.2 and iterative training.

*Our method.* We test the proposed *DECADE* described in Section 3, with the *data-dependent* local metric and the *expected alignment*, optimized in the large margin framework. We use a two layer feed forward neural network with sigmoid activation function as the deep local distance model. Each layer of the network has the same input and output size, which is the dimension of the time series $P$. For the alignment length range in *DECADE*, we set $U_l = T_{ave}, U_h = 1.5T_{ave}$, where $T_{ave}$ is average time series length in the dataset. The value of the hyper-parameters $\delta$ and ratio $\lambda$ are chosen with proper cross validation. We set the numbers of targets and imposters to be 3 and 10 respectively. Our experimental results also show that the performance of *DECADE* does not heavily depend on the value of hyperparameters if their values are in a reasonable range.

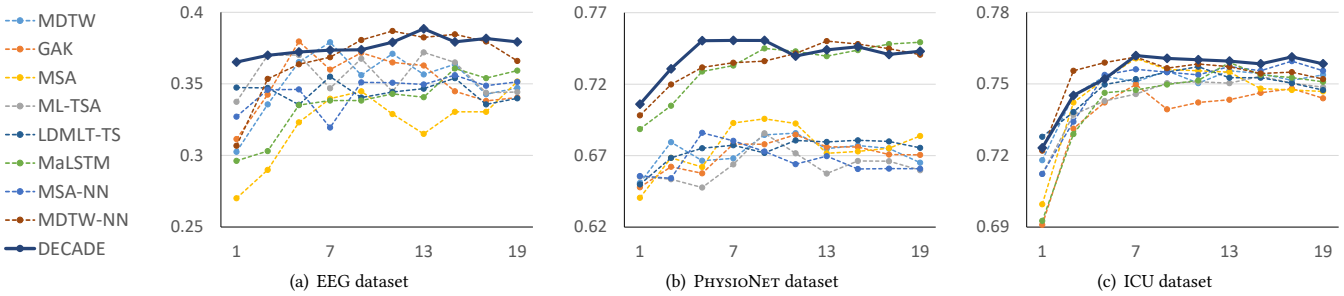
## 4.3 Experimental results

*Nearest Neighbor classification.* We evaluate all the methods in terms of classification accuracy on all the three real world datasets, with 5-fold cross validation. Table 2 shows the 1-nearest neighbor classification accuracy on all datasets, and the $k$-nearest-neighbor classification results with $k$ from 1 to 19 are shown in Figure 2 Overall *DECADE* performs the best among all the baselines on two of three datasets in terms of 1-NN classification accuracy. Moreover, in complementary experiments it outperforms the baselines across a wide variety of scenarios with different numbers of neighbors. For detailed analysis, we first compare the difference of data-independent similarities to data-dependent similarities learned from data across different datasets. We observe that on average the improvement from learning data-dependent similarities is more significant on the EEG dataset with more input dimensions. Specifically, *DECADE* improves over best data-independent similarity measures *GAK* significantly by more than 17% percent while the improvement on PhysioNet dataset with lower feature dimension is about 8%. These observations demonstrate the necessity of learning data-dependent local distance to capture high-dimension complex interactions. Next, we compare the performance of *DECADE* to similarity measures, *MSA-NN* and *MDTW-NN*, which use deep models as local distances. Our method outperformes both of them on all of the three datasets. This observation implies that a deep local distance model alone is not enough to achieve accurate similarity. The expected alignment allows *DECADE* to directly learn the local distance without iterations of finding alignments and thus resulting in better metrics. Moreover, the training of *DECADE* is more efficient since other approaches need to compute the alignments frequently. Additionally, we observe that *DECADE* outperforms *MaLSTM* on all datasets showing the difficulties of captures long-term dependence solely by LSTM. Another interesting observation is that the standard deviation of accuracy across 5 folds for *DECADE* is much smaller than that of the baselines. We attribute the robustness of our method to the expected alignments where the global distance is the average over many alignment paths of different lengths.

**Table 2: 1-nearest neighbor classification accuracy. (mean ± standard deviation)**

| Method \ Dataset | EEG | PhysioNet | ICU |
|---|---|---|---|
| *MDTW* | 0.3026 ± 0.06 | 0.6509 ± 0.05 | 0.7180 ± 0.02 |
| *GAK* | 0.3114 ± 0.05 | 0.6479 ± 0.05 | 0.6910 ± 0.03 |
| *MSA* | 0.2700 ± 0.03 | 0.6553 ± 0.05 | 0.6996 ± 0.02 |
| *ML-TSA* | 0.3375 ± 0.06 | 0.6406 ± 0.04 | 0.7123 ± 0.02 |
| *LDMLT-TS* | 0.3475 ± 0.03 | 0.6499 ± 0.04 | **0.7278 ± 0.03** |
| *MaLSTM* | 0.2963 ± 0.02 | 0.6886 ± 0.03 | 0.6926 ± 0.02 |
| *MSA-NN* | 0.3271 ± 0.05 | 0.6557 ± 0.02 | 0.7123 ± 0.02 |
| *MDTW-NN* | 0.3067 ± 0.05 | 0.6981 ± 0.02 | 0.7220 ± 0.02 |
| *DECADE* | **0.3652 ± 0.01** | **0.7060 ± 0.02** | 0.7232 ± 0.02 |

*Kernel SVM classification.* One advantage from a valid distance metric is that it can produce positive semi-definite kernels and thus be used safely in many kernel methods. While some other similarities, such as DTW, are often plugged into kernel methods in practice but have no guarantees and poor generalizations [15]. Thus, we take kernel SVM to further demonstrate the superiority or DECADE. We tested *DECADE* with *MDTW* and *GAK*, and two other baselines *LDMLT-TS* and *MDTW-NN* which give the best performance in the previous 1-nearest neighbor classifications. We build Gaussian RBF kernel with all these similarities except for *GAK*, which we use as kernel directly. As shown in Table 3, SVM with kernel built on DECADE performs the best among all SVM models.




(a) EEG dataset (b) PhysioNet dataset (c) ICU dataset

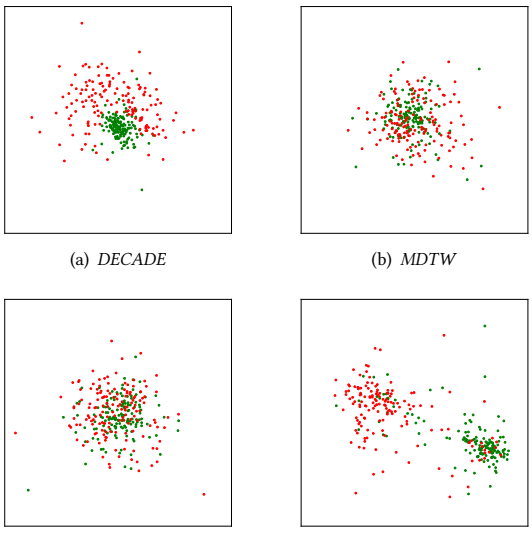
**Figure 2: Nearest neighbor classification results on real-world health care datasets. $x$-axis: number of nearest neighbors ($k$) in $k$-nearest neighbor classification; $y$-axis: classification accuracy.**

**Table 3: Gaussian RBF Kernel SVM classification accuracy. (mean ± standard deviation)**

| Method \ Dataset | EEG | PhysioNet | ICU |
|---|---|---|---|
| *SVM-MDTW* | 0.3705 ± 0.03 | 0.7155 ± 0.02 | 0.7665 ± 0.01 |
| *SVM-GAK* | 0.3658 ± 0.04 | 0.7209 ± 0.02 | 0.7468 ± 0.01 |
| *SVM-LDMLT-TS* | 0.3759 ± 0.01 | 0.7298 ± 0.02 | 0.7630 ± 0.01 |
| *SVM-MDTW-NN* | 0.3764 ± 0.03 | 0.7382 ± 0.01 | **0.7670 ± 0.01** |
| *SVM-DECADE* | **0.3974 ± 0.03** | **0.7429 ± 0.01** | **0.7670 ± 0.02** |

*Time series embedding visualization.* We visualize the 2-dimensional embedding of time series from the PhysioNet dataset in Figure 3. Similar to the visualization of synthetic dataset in Section 3.3, we apply multidimensional scaling based on their pairwise distance from *DECADE*, *MDTW* and *LDMLT-TS*. To keep the plot uncluttered, we only visualize the time series with high classification confidence. *DECADE* provides more coherent clusters of patients without in-hospitality mortality (the cluster of center points in green) when compared to *MDTW* and *LDMLT-TS*. For *MaLSTM*, though the two groups are also separated, more outliers are shown in the center of each cluster. The visualization results demonstrate that data-dependent *DECADE* can capture the complex similarity measures much more accurately. Moreover, we observe that the records of patients with in-hospitality mortality (in red) spread out much more than the rest (in green) centered in the middle, especially with the data-dependent local distance. This is indeed reasonable since records related to mortality usually have extreme or abnormal values while records of healthy patients are more similar to each other with values within a normal range, and it is also not captured by *MaLSTM*.

*Effective size of target and imposter neighbor sets.* The selection of the numbers of target and imposter neighbors is one of the key factor in determining the training cost of *DECADE*. Ideally but impractically, using all pairs of time series potentially provides the best performance with the slowest training speed. We test different numbers of target and imposter neighbors and report the $k$-nearest neighbor classification results on PhysioNet dataset in Figure 4. With only 3 targets and 10 imposters and one hidden layer we can get the best performance on this dataset. This indicates that a small subset of targets and imposters is enough for good performance, which makes the training efficient. The small number of required target and imposter samples together with the efficient sampling of the expected alignment make our *DECADE* efficient for large-scale datasets.



(a) *DECADE* (b) *MDTW* (c) *LDMLT-TS* (d) *MaLSTM*

**Figure 3: Embedding of PhysioNet dataset in 2 dimensions from *DECADE*, *MDTW*, *LDMLT-TS*, and *MaLSTM*. Red/green points refer to patient record with/without in-hospital mortality.**

*Comparisons on local metric and different alignments in DECADE.* One question regarding to the model design is that, whether the data-dependent local metric and expected alignment in DECADE- are both indispensable? Why not use the expected alignment on the raw input directly, or train the data-dependent local distance on the best alignment? On one hand, we know the expected alignment part provides a valid metric and thus several good properties and theoretical guarantees. However, taking all alignments into

**Table 4: Comparison of *MDTW* (no learned local metric, best alignment), *EA* (no learned local metric, all alignments), *MDTW-NN* (learned local metric, best alignment), and *DECADE* (learned local metric, all alignments). 1-nearest neighbor classification accuracy (mean ± standard deviation) is shown.**

| EEG | | PhysioNet | | ICU | |
|---|---|---|---|---|---|
| *MDTW* | *EA* | *MDTW* | *EA* | *MDTW* | *EA* |
| 0.3026 ± 0.06 | 0.2845 ± 0.03 | 0.6509 ± 0.05 | 0.5326 ± 0.05 | 0.7180 ± 0.02 | 0.6811 ± 0.01 |
| *MDTW-NN* | *DECADE* | *MDTW-NN* | *DECADE* | *MDTW-NN* | *DECADE* |
| 0.3067 ± 0.05 | 0.3652 ± 0.01 | 0.6981 ± 0.02 | 0.7060 ± 0.02 | 0.7220 ± 0.02 | 0.7232 ± 0.02 |

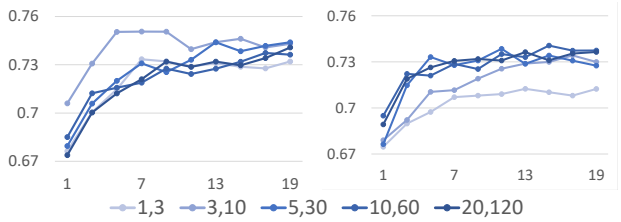



**Figure 4: Classification accuracy on PhysioNet dataset for *DECADE* with different numbers of targets and imposters with 1 sigmoid hidden layer (left) and 2 hidden layers (right). Each curve refers to a setting of (# of targets, # of imposters); $x$-axis: number of nearest neighbors ($k$) used $k$-nearest neighbor classification; $y$-axis: classification accuracy.**

consideration might not be helpful on raw input space, and thus we need deep neural networks to learn the metric data with labels to improve the quality of the metric. On the other hand, since the alignment with minimum distance is dependent from the local metric and thus is dependent from the neural networks, the objective function in the end-to-end training on the best alignment is inefficient and requires alternative updates on the neural network and the best alignment. Thus it is easy to be trapped in local optima and inefficient. In order to demonstrate this, we also tested the expected alignment itself without learned local metric, which is named as *EA*. We compare the 1-nearest neighbor results in Table 4. We can find using expected alignment only is not effective enough in practice, and combining local metric together provide the best performance.

## 5 CONCLUSION

In this paper, we propose an effective metric learning framework based on a novel global metric called *Deep ExpeCted Alignment DistancE* (DECADE) for multivariate time series data. DECADE can provide valid time series metric, learn data-dependent metric while considering temporal alignment coherently within one framework, by its two indispensable components: a novel alignment mechanism called expected alignment and a data-dependent local metric learned by deep neural networks. Our experimental results on synthetic and real world health-care datsets demonstrate that DECADE is superior among state-of-the-art time series similarity measures. The success of DECADE and its corresponding learning framework in classification tasks also indicates great potential in solving other related problems, such as multivariate time series dimension reduction and time series hashing.

## REFERENCES

[1] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 359–370.
[2] Ingwer Borg and Patrick JF Groenen. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.
[3] Marco Cuturi. 2011. Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 929–936.
[4] Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. 2007. A kernel for time series based on global alignments. In *ICASSP*, Vol. 2. II–413.
[5] Robert M Hamer and Pippa M Simpson. 2009. Last observation carried forward versus mixed models in the analysis of psychiatric clinical trials. (2009).
[6] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. 2015. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3279–3286.
[7] Ville Hautamaki, Pekka Nykanen, and Pasi Franti. 2008. Time-series clustering by approximate prototypes. In *ICPR 2008. 19th International Conference on Pattern Recognition*. 1–4.
[8] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association* 58, 301 (1963), 13–30.
[9] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*. Springer, 84–92.
[10] Paulien Hogeweg and Ben Hesper. 1984. The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of molecular evolution* 20, 2 (1984), 175–186.
[11] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. 2014. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1875–1882.
[12] David C Kale, Dian Gong, Zhengping Che, Yan Liu, Gerard Medioni, Randall Wetzel, and Patrick Ross. 2014. An Examination of Multivariate Time Series Hashing with Applications to Health Care. In *ICDM*.
[13] Eamonn Keogh and Shruti Kasetty. 2003. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery* 7, 4 (2003), 349–371.
[14] Rémi Lajugie, Damien Garreau, Francis Bach, and Sylvain Arlot. 2014. Metric Learning for Temporal Sequence Alignment. In *NIPS*. 1817–1825.
[15] Hansheng Lei and Bingyu Sun. 2007. A study on the dynamic time warping in kernel machines. In *Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on*. IEEE, 839–845.
[16] Jiangyuan Mei, Meizhu Liu, Yuan-Fang Wang, and Huijun Gao. 2016. Learning a Mahalanobis Distance-Based Dynamic Time Warping Measure for Multivariate Time Series Classification. *IEEE transactions on Cybernetics* (2016).
[17] Abdullah Mueen, Eamonn Keogh, and Neal Young. 2011. Logical-shapelets: An Expressive Primitive for Time Series Classification. In *KDD*. 1154–1162.
[18] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*.
[19] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. 1980. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 6 (1980), 623–635.
[20] Tohru Ozaki. 2012. *Time series modeling of neuroscience data*. CRC Press.
[21] Wenjie Pei, David MJ Tax, and Laurens van der Maaten. 2016. Modeling Time Series Similarity with Siamese Recurrent Networks. *arXiv preprint arXiv:1603.04713* (2016).
[22] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching

and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In *KDD*.

[23] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26, 1 (1978), 43–49.

[24] Stan Salvador and Philip Chan. 2007. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[25] Christopher L Sistrom, Pragya A Dang, Jeffrey B Weilburg, Keith J Dreyer, Daniel I Rosenthal, and James H Thrall. 2009. Effect of Computerized Order Entry with Integrated Decision Support on the Growth of Outpatient Procedure Volumes: Seven-year Time Series Analysis 1. *Radiology* 251, 1 (2009), 147–155.

[26] Li Wei and Eamonn Keogh. 2006. Semi-supervised Time Series Classification. In *KDD*.

[27] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10 (2009), 207–244.

[28] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. 2006. Fast Time Series Classification Using Numerosity Reduction. In *ICML*. 1033–1040.

[29] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. 2014. Deep metric learning for person re-identification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 34–39.