IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 7, 2020, accepted January 18, 2021, date of publication January 22, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053703

Deep Multiple Metric Learning for Time Series Classification

ZHI CHEN[1], YONGGUO LIU[1], JIAJING ZHU[1], YUN ZHANG[1], QIAOQIN LI[1], RONGJIANG JIN[2], AND XIA HE[3]

[1]Knowledge and Data Engineering Laboratory of Chinese Medicine, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]College of Health Preservation and Rehabilitation, Chengdu University of Traditional Chinese Medicine, Chengdu 610075, China
[3]Sichuan 81 Rehabilitation Center, Chengdu 610035, China

Corresponding author: Yongguo Liu (liuyg@uestc.edu.cn)

This research was supported in part by the National Key R&D Program of China under grant 2019YFC1710300 and SQ2018YFC200065-02, and the Sichuan Science and Technology Program under grants 2021YJ0184, 2020YFS0283, 2020YFS0302 and 2019YFS0019.

ABSTRACT Effective distance metric plays an important role in time series classification. Metric learning, which aims to learn a data-adaptive distance metric to measure the distance among samples, has achieved promising results on time series classification. However, most existing approaches focus on learning a single linear metric, which is unsuitable for nonlinear relationships and heterogeneous datasets with locality information. Besides, the hard samples in the training set account for only a small part, which may fail to characterize the global geometry of the metric embedding space. In this paper, we propose a novel deep multiple metric learning (DMML) method for time series classification. DMML contains a convolutional network component to extract nonlinear features of time series. For exploiting locality information, the last feature layer of the convolutional network is divided into several nonoverlapping groups and a separate metric learner is built on each group to get multiple metrics. In order to reduce the correlations among learners and facilitate robust metric learning, we design an adversarial negative generator to synthesize different hard negative complements for different metric learners. Moreover, an auxiliary loss is introduced to increase the robustness of DMML for the magnitude of distance. Extensive experiments on UCR datasets demonstrate the effectiveness of DMML for time series classification.

INDEX TERMS Adversarial training, deep learning, metric learning, time series classification.

I. INTRODUCTION

Since that time series data are generated in a wide range of real-life domains, including healthcare [1], finance [2], and meteorological [3], time series research has attracted significant interests within the data mining community. The pervasiveness of time series inspires machine learning techniques for time series analysis, such as classification and forecasting. In this paper, we mainly focus on time series classification.

Most existing time series classification methods focus on designing an effective distance metric among series and classify a time series to the same class as its nearest time series according to the distance metric [4]–[7]. These methods usually use hand-crafted distance metric and perform the same distance function on all the tasks ignoring the differences in data distribution. However, the distance metric should be task-specific because different datasets usually subject to variable distributions [8]. To learn a data-adaptive distance metric from the dataset, several metric learning-based methods that aim to enlarge the similarity of each positive time series pair and reduce that of each negative time series pair have been proposed for time series classification [9]–[11]. For instance, Do *et al.* [12] proposed to learn a distance metric by combining several modalities at multiple temporal scales for an effective k nearest neighbors classification. Although these methods provide competitive or acceptable performance, they usually learn a linear distance metric and cannot capture the nonlinear manifold of time series [8].

More recently, several deep metric learning approaches have been proposed to address the nonlinear problem by learning a nonlinear embedding with deep neural networks [13] and yielded impressive performance gains on the tasks including feature matching [14], classification [15] and

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

collaborative filtering [16]. The goal of deep metric learning is to build an embedding space on the deep feature representation to capture the semantic similarity of data [17]. The deep feature representation and semantically meaningful embedding are jointly learned by a neural network model. Despite the impressive results achieved by the current researches, there are still some limitations:

- Existing methods only learn a single distance metric, which is difficult to adapt for heterogeneous datasets with multiple relationships and may not be able to handle the data varying locally. Recently, the investigations on local distance metric learning have considered locality specific approaches, and consequently multiple metrics are learned to capture multiple relationships [8]. However, these methods seldom exploit complementary information of metrics, which is of great importance for the performance of local distance metric learning.
- For most metric learning methods, the training procedure is to minimize a loss function weighted by the selected samples, and hence, the selected samples play an import role for the performance of metric learning [18]. However, vast majority of samples in the training set may satisfy the constraints imposed by the loss function and produce gradients close to zero, providing little supervision information for the training model [19]. The informative hard samples are inadequate to characterize the global geometry of the embedding space comprehensively.
- Moreover, the loss of metric learning is sensitive to the magnitude of distance. Traditional methods that only use the metric loss function can optimize the loss by shrinking the magnitude of distance, which is meaningless and may degrade classification performance. For example, as shown in Fig. 1(a), there are four classes in the training set before optimizing the metric learning loss. Obviously, each sample in class 3 and class 4 shares the same label as its nearest neighbor and we can correctly classify the samples in class 3 and class 4 based on 1 nearest neighbor classifier. Besides, the distance relationship between the samples of class 1 and class 2 is incorrect. The metric learning loss can be optimized by shrinking the distance between all samples in horizontal direction, as shown in Fig. 1(b). However, this update not only does not benefit the classification for the samples in class 1 and class 2 but destroys the classification for the samples in class 3 and class 4.

To address these problems, we propose an effective deep multiple metric learning (DMML) model for time series classification. For capturing the nonlinear manifold of time series, a deep convolutional neural network is designed to project time series into a nonlinear feature space. For exploiting locality information, the nonlinear feature is divided into multiple nonoverlapping groups and a metric learner is established for each group to learn multiple distance metrics. To reduce the correlations between metrics and facilitate robust metric learning, a hard negative mining strategy is proposed to synthesize hard samples as the complementary training data for successive groups. The metric learner in successive group distinguishes both original training samples and the adversarial samples that generated in previous group. The metric learning and hard negative generating are simultaneously trained in an end to end fashion. Besides, we introduce an auxiliary loss to increase the robustness for the magnitude of distance. The concatenation of all metric embeddings serves as the input for k nearest neighbor classifier.

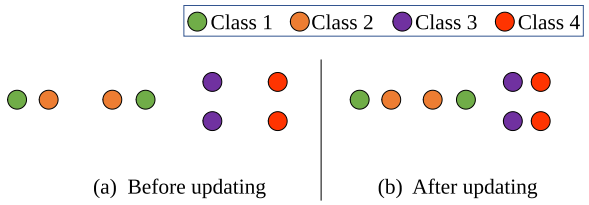


FIGURE 1. Illustration of metric degradation.

This paper's main contributions are summarized as:

- We propose a novel framework DMML, which is able to learn multiple complementary distance metrics and capture the nonlinear relationships for time series classification.
- We design an adversarial hard negative mining strategy to reduce the correlations between metrics and facilitate robust metric learning.
- We introduce an auxiliary loss to increase the robustness for the magnitude of distance.
- Comprehensive experimental results show the effectiveness of our proposed approach for time series classification compared with different types of methods.

The rest of the paper is organized as follows. In Section II, we briefly review previous studies on time series classification and metric learning. In Section III, we introduce the details of the proposed method, including deep multiple metric learning, adversarial hard negative mining and auxiliary loss. In Section IV, our proposed DMML method is evaluated and compared with the state-of-the-art methods. Section V concludes this paper.

II. RELATED WORK

In this section, we briefly review two related topics: time series classification and metric learning.

A. TIME SERIES CLASSIFICATION

Time series classification is an important topic in machine learning research. There are four representative types of techniques for time series classification: (1) shapelet-based methods, (2) feature-based methods, (3) deep learning-based methods, and (4) distance-based methods.

Shapelet-based methods extract short discriminative series segments called shapelets rather than the full series to offer interpretable results [20]–[23]. These methods must scan a large pool of subsequences, leading to a time-consuming process. Different with searching subsequences on existing

sequences, shapelet learning adopts an optimization scheme to learn discriminative subsequences from raw numeric data [24]–[27], which improves the accuracy significantly. For example, SAX-VFSEQL [27] and efficient learning interpretable shapelets (ELIS) [24] generate shapelet candidates by aggregating approximation and adjust the shape of shapelets using shapelet learning.

Feature-based methods extract discriminative features for time series classification. For instance, time series classification based on a bag-of-features representation (TSBF) [28] computes the features of random subsequences to handle warping. Time series forest (TSF) [29] adopts a random feature sampling strategy to reduce the computational complexity. Highly comparative feature (HCF) [30] constructs feature-based representations of time series using highly comparative method.

Deep learning-based methods extract nonlinear features from time series by neural networks. For example, Lin and Runger [31] presented a group-constrained convolutional recurrent neural network to model time series data. Ma *et al.* [32] simplified the learning process of echo state network through spatio-temporal aggregation operation and orthogonal function basis expansion. Wang *et al.* [33] proposed a time series classification algorithm based on echo state network and adaptive differential evolution. Chen *et al.* [10] conducted metric learning in the model space of echo state network.

Distance-based methods classify a time series to a certain category based on the distance to the tagged samples. Commonly used distance metric include dynamic time warping (DTW) distance [5], longest common subsequence (LCS) distance [6], spatial assembling distance (SAD) [7], etc. DTW aims to find an optimal warping path between two series to deal with the problem of phase aberration and regards the distance between the warping path as the distance between two series. LCS exploits the time series distance measurement based on derivatives. SAD is proposed to handle shifting and scaling in both temporal and amplitude dimensions. However, these distance metrics ignore the differences in data distribution and cannot capture the nonlinear manifold of time series [8]. Unlike existing methods that define the same linear distance metric for all datasets, the proposed DMML aims to learn multiple nonlinear distance metrics by utilizing the nonlinear feature extraction capability of neural networks in a data-adaptive manner.

B. METRIC LEARNING

Metric learning aims to learn an effective metric to measure the distance of the input pairs. The existing metric learning methods can be classified into linear metric learning methods, kernel trick-based methods, and deep learning-based methods based on literature [8], [18], [19].

Traditional metric learning methods seek a linear Mahalanobis distance [34]–[38]. For instance, information theoretic metric learning (ITML) [34] formulates the problem of metric learning as a constrained optimization task by minimizing the differential relative entropy. Large margin nearest neighbor (LMNN) [35] learns a linear transformation under which the k nearest neighbors of each data point sharing the same label. Pairwise-constrained component analysis (PCCA) [36] learns a projection that projects similar pairs inside a ball while dissimilar pairs are pushed away by gradient descent method. However, these methods cannot explicitly obtain the nonlinear mappings [39].

Kernel trick-based methods [40] are proposed to address the problem of nonlinear correlations of samples. Yeung and Chang [41] formulated the metric learning problem as an optimization problem for kernel learning. Wang *et al.* [42] generalized popular metric learning methods by a kernel classification framework. To handle multilabel learning and tasks with continuous decision values, Zhu *et al.* [40] formulated metric learning as a kernel regression problem. However, it is difficult and empirical to choose a proper kernel with flexible expression power.

With the ability of learning hierarchical nonlinear transformations, deep learning has been studied to address both the nonlinear and scalability problems simultaneously. Most of the deep metric learning methods are proposed for visual tasks, such as person re-identification [43], fine-grained visual categorisation [17], kinship verification [44] and images retrieval [45]. Song *et al.* [45] proposed lifted structured embedding to take full advantage of training batches. Duan *et al.* [19] adopted a hard negative generator to generate synthetic hard negatives from the observed negative samples. However, a single distance metric is often unable to accurately handle the task in which the data are multimodal or the decision boundary is complex [46]. Differently, DMML aims to train multiple distance metrics and generate hard negatives based on current metric for successive metric. In this way, different metrics are trained with different samples and complement each other well. Duan *et al.* [8] developed a deep localized metric learning (DLML) model to cluster all samples into multiple groups and construct a deep neural network to learn a distance metric based on each group. However, constructing multiple deep neural networks is computationally expensive. DMML divides the feature vector of a network into multiple groups for constructing multiple metric learners, which reduces the computational consumption. Moreover, each metric learner is trained with a subset of the entire dataset in DLML, which may not fully exploit the information buried in all samples. Differently, each metric learner in the proposed DMML framework is trained with all training samples and synthetic hard samples generated in the previous metric learner, which makes full use of all training samples.

In the filed of time series classification, Do *et al.* [12] proposed to learn a distance metric by combining several modalities at multiple temporal scales for an effective k nearest neighbors classification. Gong *et al.* [47] utilized a multiobjective model-metric learning framework based on recurrent network. However, these approaches focus on learning a single distance metric, which is unsuitable for

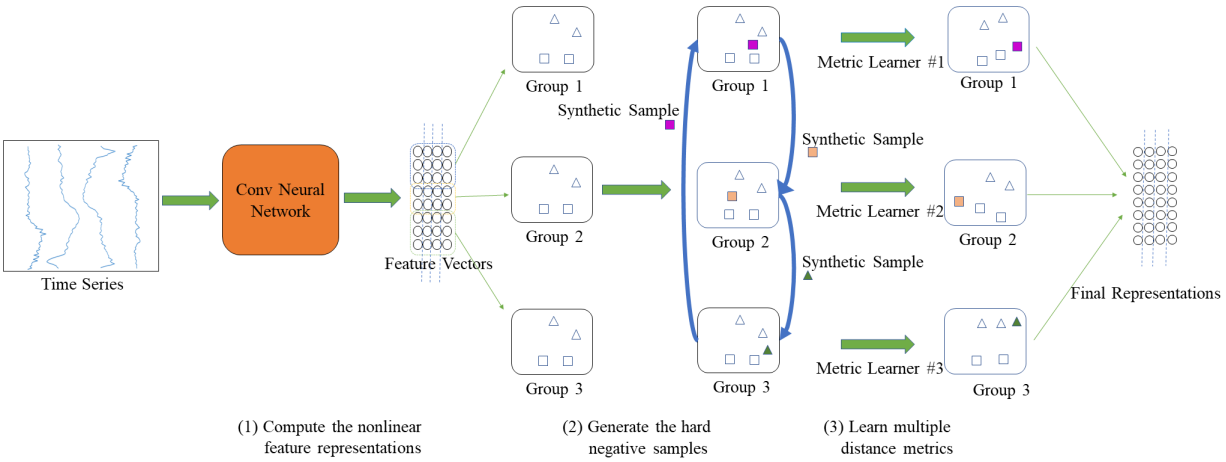


FIGURE 2. Illustration of the proposed DMML method. Given a set of time series, we first construct a convolutional network to get the feature vectors and divide the feature vectors into multiple nonoverlapping groups (step 1). Then we synthesize hard negatives for successive metric learners by the hard negative generator (step 2). We conduct metric learning on each group and get the final embedding by concatenating the embeddings on all metric learners (step 3). The hard negative generating and metric learning is trained in an adversarial manner to learn multiple distance metrics.

heterogeneous datasets with locality information. Different from previous methods that only learn a single distance metric based on original training samples, we aim to learn multiple distance metrics to exploit locality information. Moreover, a hard negative mining strategy is proposed to synthesize hard samples as the complementary training data for reducing the correlations between metrics and facilitate robust metric learning.

III. PROPOSED APPROACH

In this section, DMML is described in detail. We first introduce the linear metric learning-based time series classification. The convolutional network-based deep multiple metric learning is then explicated. After that, the hard negative generator is detailed. Finally, the auxiliary loss is presented. Fig. 2 illustrates the pipeline of DMML.

A. LINEAR METRIC LEARNING

Let $X = \{(x_i, y_i)\}_{i=1}^{N}$ be a set of N time series, where x_i and y_i are the i-th time series and its corresponding label, respectively. Given an unclassified time series, we assign to it the label of the nearest neighbor among the training set according to a distance metric. Therefore, we aim to learn a distance metric under that the distances between dissimilar samples are enlarged while the distances between similar ones are reduced.

Most existing metric learning methods focus on learning the Mahalanobis distance [34]–[38]. Given samples x_i and x_j, where $x_i, x_j \in \mathbb{R}^d$ and d is the dimension of the feature space (For a given time series, its dimension of the feature space is usually the length of time series), Mahalanobis distance $D_M(x_i, x_j)$ is defined as

$$D_M(x_i, x_j) = (x_i - x_j)^\top M (x_i - x_j). \tag{1}$$

We denote $D_M(x_i, x_j)$ as $D_{i,j}$ for simplicity. The symmetric positive definite (SPD) matrix $M \in \mathbb{R}^{d\times d}$ called Mahalanobis matrix is learned by an algorithm to fit the distance reflected by the training data. The SPD Mahalanobis matrix can be decomposed as $M = W^\top W$, where $W \in \mathbb{R}^{m\times d}$ is the transformation matrix and m is embedding size. Then the Mahalanobis distance between x_i and x_j can be rewritten as

$$\begin{aligned} D_{i,j} &= (x_i - x_j)^\top M (x_i - x_j) \\ &= (x_i - x_j)^\top W^\top W (x_i - x_j) \\ &= (Wx_i - Wx_j)^\top (Wx_i - Wx_j) \\ &= ||(Wx_i - Wx_j)||^2. \end{aligned} \tag{2}$$

It can be seen that learning a Mahalanobis distance metric is equivalent to seeking a linear transformation W which projects a sample into a new space. In this space, dissimilar samples should be far apart from each other, whereas similar samples should be close to each other. Therefore, a large number of models are proposed to either directly learn Mahalanobis matrix M or indirectly learn transformation matrix W.

Then, we present the loss function for training the distance metric. The loss function adopted in this work is the lifted structure loss [45]. The structured loss encourages the smallest distance between the samples in a positive pair and their negatives is larger than a margin, which is defined as

$$\begin{aligned} \mathcal{L}_{struc}(X) &= \frac{1}{2|P|} \sum_{(x_i,x_j)\in P} \max(0, l(x_i, x_j))^2. \\ l(x_i, x_j) &= \max\left(\max_{(x_i,x_q)\in N} \alpha - D_{i,q}, \max_{(x_j,x_l)\in N} \alpha - D_{j,l} \right) \\ &\quad + D_{i,j}, \end{aligned} \tag{3}$$

where P and N are the set of positive pairs and the set of negative pairs in the training set, respectively, α is the margin. However, this loss is non-smooth and requires all sample pairs several times. Lifted structured loss address these challenges

by a smooth upper bound on the function:

$$\mathcal{L}_{lifted}(X) = \frac{1}{2|P|}\sum_{(x_i,x_j)\in P}\max(0, l(x_i,x_j))^2.$$
$$l(x_i,x_j) = \log\Big(\sum_{(x_i,x_q)\in N}\exp\{\alpha - D_{i,q}\} + \sum_{(x_j,x_l)\in N}\exp\{\alpha - D_{j,l}\}\Big) + D_{i,j},\ \textit{if}\ (x_i,x_j)\in P, \tag{4}$$

Parameter α is set to 1 following Song *et al.* [45].

B. DEEP MULTIPLE METRIC LEARNING

Traditional metric learning methods only seek a linear metric and cannot well exploit the nonlinear manifold of time series. In DMML, we construct a convolutional neural network to compute the nonlinear feature representations of time series to overcome this limitation. The convolutional neural network can be viewed as a nonlinear function ϕ that maps time series x_i into an feature representation $x_i' = \phi(x_i)$, $x_i' \in \mathbb{R}^r$, where r is the size of feature representation. As a result, the distance between two time series x_i and x_j passing through the network can be written as

$$\begin{aligned} D_M(i,j) &= (\phi(x_i) - \phi(x_j))^\top M(\phi(x_i) - \phi(x_j)) \\ &= ||(W\phi(x_i) - W\phi(x_j))||^2. \end{aligned} \tag{5}$$

This lets us incorporate metric M into a deep net in the form of a fully connected layer before a loss layer. However, learning a holistic distance metric over the input space is not able to fully capture the relationships between input pairs. Inspired by the fact that localized metric learning approaches learn a set of local metrics, we aim at learning multiple complementary distance metrics. For deep metric learning, one way is to train several convolutional networks and learn a metric on each network. However, training multiple convolutional networks is computationally expensive. In this work, the shared computation of convolution [48] is used to get an elegant and effective solution. Specifically, the feature vector $\phi(x_i)$ is divided into K nonoverlapping subfeature groups:

$$\phi_k(x_i) = \phi(x_i)[\pi_k : \pi_{(k+1)}], \tag{6}$$

where $\phi_k(x_i)$ is the k-th subfeature group, π_k is the split point between groups $k-1$ and k, and all groups have the same size. Then, a separate metric learner is built on each group. The loss function for a positive pair on the k-th group is

$$l_k(x_i,x_j) = \log\Big(\sum_{(x_i,x_q)\in N}\exp\{\alpha - D^k_{i,q}\} + \sum_{(x_j,x_l)\in N}\exp\{\alpha - D^k_{j,l}\}\Big) + D^k_{i,j},\ \textit{if}\ (x_i,x_j)\in P. \tag{7}$$

$D^k_{i,j}$ is the distance between x_i and x_j on the k-th group, which is defined as

$$D^k_{i,j} = ||(W_k\phi_k(x_i) - W_k\phi_k(x_j))||^2, \tag{8}$$

where W_k is the transformation matrix of the k-th metric learner. The loss function on the k-th group in the training dataset X is defined as

$$\mathcal{L}_k(X) = \frac{1}{2|P|}\sum_{x_i,x_j\in P}\max(0, l_k(x_i,x_j))^2. \tag{9}$$

Then the global loss function for deep multiple metric learning is given as

$$\mathcal{L}_{lifted}(X) = \sum_k \mathcal{L}_k(X). \tag{10}$$

After learning all metrics, we concatenate them to get the final distance metric. Our deep multiple metric learning framework is constructed based on a shared convolutional network. Therefore, all distance metrics share the same underlying feature representation and the computational consumption is reduced.

C. HARD NEGATIVE MINING

If we only optimize the global loss function on the training set, the metrics with high correlation are then learned, resulting in no performance improvements at all [49]. Besides, due to different samplings between training set and test set, the trained models often fail to learn a reliable metric on the ambiguous test pairs. Therefore, it is not desirable to train all metric learners on the same data directly.

To overcome these limitations, we introduce the adversarial hard negative generator to synthesize different hard negative complements for different metric learners. Our goal is to generate potential hard negatives based on original training data and current metric to provide synthetic complements for successive metric learner. The generator aims to synthesize negative samples that are confused for current metric learner. The metric learner in the successive group distinguishes both original training samples and the adversarial samples generated in the previous group. Different metrics are trained with different samples, which reduces the correlations among learners.

We add perturbations to existing samples to get the adversarial samples. Specifically, we construct a perturbation generator based on original training data and the k-th metric:

$$r_{k,i} = \epsilon\frac{g_k(x_i, x_i^-, W_k)}{||g_k(x_i, x_i^-, W_k)||}, \tag{11}$$

where x_i^- is the sample with different label from x_i, $r_{k,i}$ is the synthetic perturbation, g_k is the perturbation generator, and ϵ is the parameter that controls the magnitude of perturbation. We concatenate x_i and x_i^- as the input of generator g_k. The synthetic adversarial sample is

$$\tilde{x}^-_{k,i} = x_i^- + r_{k,i}. \tag{12}$$

Since that only a small perturbation is applied to existing sample x_i^- in the input space, $\tilde{x}^-_{k,i}$ should share the same label with x_i^-. The goal of generator is to synthesize negative samples difficult for the current metric to classify in

the embedding space. The loss function of the generator is formulated as

$$\begin{aligned}\mathcal{L}_{gen} &= \sum_{k} \sum_{(x_i, x_i^-) \in X} \mathcal{L}_{pn} - \mathcal{L}_{nn} \\ &= \sum_{k} \sum_{(x_i, x_i^-) \in X} \left(\left\| W_k \phi\left(x_i\right) - W_k \phi\left(\tilde{x}_{k,i}^-\right) \right\|^2 \right. \\ &\quad \left. - \left\| W_k \phi\left(x_i^-\right) - W_k \phi\left(\tilde{x}_{k,i}^-\right) \right\|^2 \right)\end{aligned} \tag{13}$$

The goal of $\mathcal{L}_{pn}$ is to make the synthetic sample close to the anchor x_i in the embedding space. The second term $\mathcal{L}_{nn}$ aims to make the synthetic sample keep away from the negative sample x_i^- in the embedding space. The generator with $\mathcal{L}_{gen}$ is able to synthesize negative samples that would be misclassified by the learned metric. The generator used in this work is a convolutional network with three fully connected layers of output dimensions 64, 128 and 64, and a fully connected layer of which the dimension is the length of the input time series.

The metric leaner in the successive group is trained on both original training samples and the adversarial samples generated in previous group. The loss function is then formulated as

$$\mathcal{L}_{lifed}(X) = \sum_{k} \mathcal{L}_k(X, \tilde{X}_{k-1}), \tag{14}$$

where $\tilde{X}_{k-1}$ is the set of synthetic samples generated in previous group. The metric learning and the hard negative generating are trained with the following loss function simultaneously

$$\mathcal{L}_{metric} = \mathcal{L}_{lifted} + \mathcal{L}_{gen}. \tag{15}$$

Hard negative mining has two advantages. First, the procedure of adversarial training enhances the ability of metrics to address potential unobserved hard negatives [50]. We follow the idea of adversarial training to generate ambiguous but critical data as important complements to existing samples. Metrics in successive groups discriminate the confusing unseen adversarial samples to enhance the discriminative power. Second, training different metrics with different samples reduces the correlations among metrics. The synthetic samples that confuse the current metric are a part of the training data of the next learner. Different metrics focus on different samples and capture different local specificities.

D. AUXILIARY LOSS

For most metric learning methods, the loss is a function of distance, which is sensitive to the distance magnitude. For example, the lifted structure loss is composed of the distance between similar pairs and that between dissimilar pairs. It means that we can shrink the distance magnitude to reduce the loss. However, it is meaningless for classification and might result in some unexpected results as discussed in Section I. We call this phenomenon as metric degradation.

In this paper, we incorporate an auxiliary loss term to avoid the problem of metric degradation. Specifically, a softmax layer is added on the top of the feature representation to predict the label of input time series. Thus, the sofltmax loss function is added:

$$\mathcal{L} = \mathcal{L}_{metric} + \lambda \mathcal{L}_{aux}, \tag{16}$$

where λ is a balance factor and $\mathcal{L}_{aux}$ is the auxiliary softmax loss function. The softmax loss ensures that the optimizing of DMML is meaningful to classification to avoid metric degradation. We show the main algorithm of DMML in Algorithm 1.

Algorithm 1 DMML

Input: Training set X, number of groups K, embedding size m, parameters λ and ϵ, and iteration numbers T.

Output: Parameters of deep metric learning θ_{metric}, and parameters of the hard negative generator θ_{gen}.

```
Initialize parameters θ_metric and θ_gen.
for t = 1 to T do
    Compute nonlinear feature representation φ(x_i) of input time series x_i.
    Divide φ(x_i) into K subfeature groups using Eq. 6.
    for k = 1 to K do
        Produce hard negative sample x̃⁻_{k,i} using Eq. 12 for x_i.
        Compute nonlinear feature representation φ(x̃⁻_{k,i}) of φ(x̃⁻_{k,i}).
    end for
    Jointly optimize θ_metric and θ_gen using Eq. 16 with BackPropagation.
end for
return θ_metric and θ_gen.
```

IV. EXPERIMENTS

In this section, we first describe the implementation details. Then, we discuss how framework components contribute to DMML. Next, we test several key parameters of DMML. Finally, we compare DMML with state-of-the-art time series classification methods.

The experiments are conducted on some widely used benchmark datasets from the ''UCR Time Series Data Mining Archive'' [51]. Each dataset has been divided into a training set and a test set by the provider. The testbed provides diverse characteristics such as the number of classes, the number of samples and the length of series so as to enable a comprehensive evaluation. The ablation study and parameter analysis are conducted on 6 datasets that exhibit various characteristics for fair comparison, i.e., ECGFiveDays, FISH, MoteStrain, SonyAIBORobotSurface, SonyAIBORobotSurfaceII, and SwedishLeaf.

TABLE 1. Classification accuracy (%) comparison with different ϵ and $DMML_{-adv}$.

Dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-1}$	$\epsilon = 10^{0}$	DMML_{-adv}
ECGFiveDays	95.5	98.8	97.8	**98.9**	97.3	92.8
FISH	94.7	94.7	**95.3**	94.7	93.0	94.7
MoteStrain	89.3	**92.0**	88.7	86.1	84.5	**92.0**
SonyAIBORobotSurface	97.0	**97.1**	96.6	96.2	95.2	96.4
SonyAIBORobotSurfaceII	93.2	**95.0**	92.3	92.5	90.5	87.9
SwedishLeaf	94.3	94.0	**94.6**	94.0	93.4	92.2

TABLE 2. Classification accuracy (%) comparison with different λ, $DMML_{-a}$, and $DMML_{-m}$.

Dataset	DMML_{-a}	$\lambda = 10^{0}$	$\lambda = 10^{1}$	$\lambda = 10^{2}$	$\lambda = 10^{3}$	DMML_{-m}
ECGFiveDays	98.0	98.6	**99.4**	97.8	99.1	82.8
FISH	92.3	91.7	94.7	**95.3**	94.1	52.5
MoteStrain	87.2	87.9	**90.9**	88.7	90.1	71.6
SonyAIBORobotSurface	96.2	95.7	95.2	**96.6**	94.6	63.2
SonyAIBORobotSurfaceII	95.2	95.0	**96.3**	92.3	92.5	80.5
SwedishLeaf	79.5	91.2	94.3	**94.6**	94.2	68.3

TABLE 3. Classification accuracy (%) comparison with different K.

Dataset	$\text{DMML}_s(K=1)$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$
ECGFiveDays	96.9	98.7	97.8	**99.0**	96.9	96.3
FISH	93.0	94.1	**95.3**	94.1	94.7	92.5
MoteStrain	89.1	87.6	88.7	87.1	**90.7**	87.5
SonyAIBORobotSurface	96.6	96.9	96.6	**97.5**	97.4	96.1
SonyAIBORobotSurfaceII	89.3	91.2	92.3	**94.3**	90.1	92.7
SwedishLeaf	92.9	94.3	94.6	94.1	**95.4**	93.7
HHAR	75.2	76.2	74.2	78.4	**79.4**	77.5

A. IMPLEMENTATION DETAILS

All experiments are conducted on a computer with an Intel Xeon(R) Gold 5122 3.60-GHz CPU, 64-GB RAM, and a GeForce GTX 1080-Ti 11G graphics card. We implement DMML with PyTorch package.

The architecture of the deep network is recommended by Wang *et al.* [52]. The network is a residual network with 9 convolutional layers and a global average pooling layer, which is composed of 3 residual blocks. For each block, the lengths of the filters are set as 8, 5, and 3, respectively, and the number of filters is 128 for all convolutions. The following global average pooling layer averages the series across the time dimension. Then the global average pooling layer is followed by a softmax layer and a metric learning layer.

B. ABLATION STUDY

In this section, we discuss how framework components contribute to DMML through quantitative component-wise evaluation.

1) EFFECT OF HARD NEGATIVE MINING

In this group of experiments, we analyze the contribution of hard negative mining. The model without hard negative mining is denoted as $DMML_{-adv}$. We compare DMML with $DMML_{-adv}$. Parameters m, K, and λ are fixed to 512, 3, and 100, respectively. As shown in Table 1, DMML outperforms $DMML_{-adv}$ for most datasets. The reason is that hard negative mining generates synthetic hard negatives for different metric learners and exploits more information from a limited training set than conventional methods that only exploit the observed negatives in their original form, increasing the complementarity of metrics. With adequate and diversiform synthetic hard negatives, the final distance metric presents strong robustness.

2) EFFECT OF AUXILIARY LOSS

We show the effectiveness of our auxiliary loss. The embedding size m, the number of groups K, and the perturbation parameter ϵ are set to 512, 3 and 0.01, respectively. We first set $\lambda = 0$ to abandon the auxiliary loss, and the derived model is denoted as $DMML_{-a}$. Similarly, we cancel the metric learning loss to freeze the metric learning module and denote the derived model as $DMML_{-m}$. We show the results of different model settings in Table 2. To show the convergence of different model settings, the corresponding learning curves are shown in Fig. 3. We can make some interesting observations from these results. 1) The optimal DMML model outperforms $DMML_{-a}$, which shows that we can improve the performance of DMML by including an auxiliary loss during training. 2) The optimal DMML model outperforms $DMML_{-m}$, which shows that the metric learning module in DMML is effective. 3) The absence of metric learning loss results in the fluctuation of accuracy curves and a dramatic performance reduction, which demonstrates that if we only consider the auxiliary loss during training, it is difficult to convergence.

3) EFFECT OF MULTIPLE METRIC LEARNING

In this part, we illustrate the effect of multiple metric learning. We implement a variant of DMML named $DMML_s$ which only learns a single distance metric. The performance of the variant ($K = 1$) and DMML ($K >= 2$) are shown in Table 3. As can been seen, DMML achieves better results than the single metric learning method. The reason is that DMML exploits multiple complementary local specificities to improve the classification performance with multiple distance metrics and generated hard negatives. To show the effectiveness of DMML on heterogeneous dataset, we conduct experiments on Heterogeneity Human Activity Recognition (HHAR) dataset [53]. HHAR dataset is gathered with various device models and use-scenarios to reflect sensing heterogeneity existing in real deployments. The dataset contains 6 activities, including biking, sitting, standing, walking, stair up and stair down) collected from 9 users using smartphones and smartwatches. These devices varied in the supported maximum sampling rate which varies from 25Hz to 200Hz. Moreover, there exists different accelerometer biases and gains among different devices. These heterogeneities in the dataset increase the intra-class distance while reduce the inter-class distance. In our experiment, accelerometer measurements are model inputs, while activities are used as labels. We unify the data with different sampling rates to 10Hz using downsampling and segment the data into 5 seconds samples. We randomly select 70% samples as the training set and the rest are used as the test set. The results of different methods are presented in Table 3. As shown in Table 3, we can find that DMML achieves better results

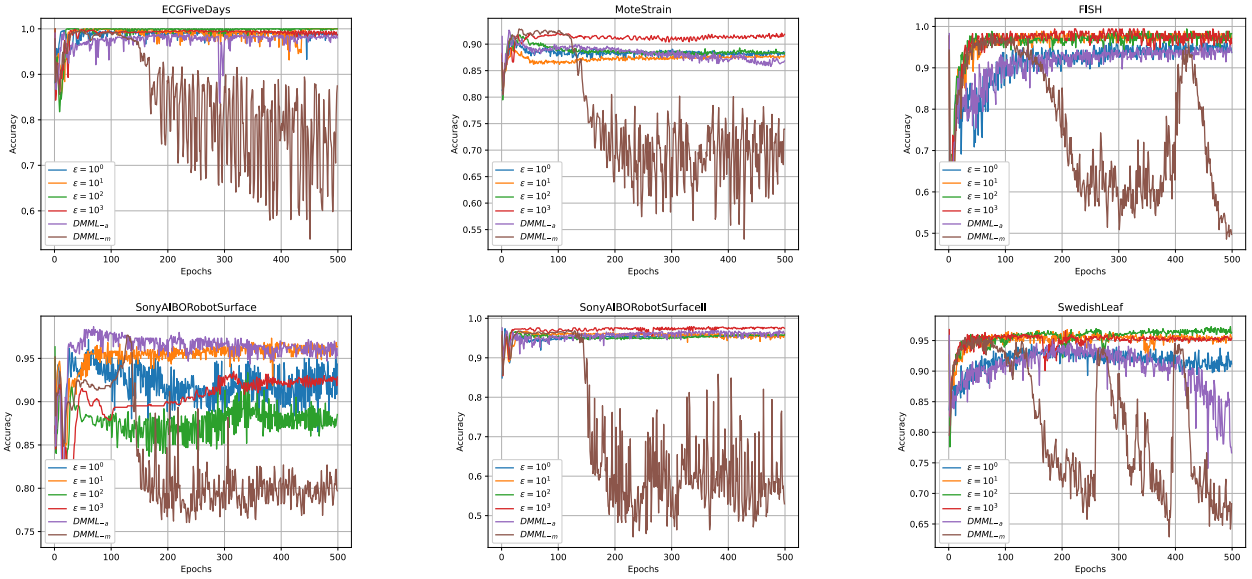


FIGURE 3. Classification accuracy comparison with different λ, $DMML_{-a}$, and $DMML_{-m}$.

than $DMML_s$ on HHAR dataset. The results show that our DMML method that learns multiple distance metrics to exploit the local specificities is beneficial for improving the performance on the task of heterogeneity human activity recognition.

C. PARAMETER ANALYSIS

In the proposed algorithm, several parameters may influence the time series classification performance, including the number of groups K, embedding size m, perturbation parameter ϵ, and balance factor λ. We conduct experiments to illustrate the effects of these parameters on DMML.

1) IMPACT OF NUMBER OF GROUPS AND EMBEDDING SIZE

We first illustrate the effects of K and m while fixing ϵ and λ at 0.01 and 100, respectively. To evaluate the impact of K, we fix the embedding size as 512. Then we run our model with $K = \{1, 2, 3, 4, 5, 6\}$. For $K = 1$, the model degenerates into a single metric learning model. Table 3 shows the variations in the accuracy of different numbers of groups. We find that the optimal number of groups for DMML is in the range of [4, 5]. Besides, we see that when K is small, the classification accuracy of the proposed method is low. The reason for this phenomenon is that, for a limited number of groups, a small number of metrics are learned, which is not able to fully capture the relationships between input pairs. With more groups, the final representation captures more local specificities of data. However, a large number of groups mean a lot of adversarial samples need to be generated, increasing the computational complexity and memory cost during the training. These observations show that our multiple metric learning with a proper K can exploit multiple complementary local specificities to improve the classification performance.

TABLE 4. Classification accuracy (%) comparison with different m.

Dataset	$m = 64$	$m = 128$	$m = 256$	$m = 512$	$m = 1024$
ECGFiveDays	98.1	**98.9**	96.7	97.8	97.5
FISH	93.5	93.2	94.7	**95.3**	94.7
MoteStrain	88.1	88.1	**91.6**	88.7	88.0
SonyAIBORobotSurface	95.8	**96.6**	96.1	**96.6**	95.9
SonyAIBORobotSurfaceII	93.1	**93.2**	92.8	92.3	93.0
SwedishLeaf	93.2	94.2	93.8	**94.6**	93.4

Then we analyze the impact of m. In the experiments, we fix K at 3 and run our model with $m = \{64, 128, 256, 512, 1024\}$, respectively. The results of DMML with different m are shown in Table 4. We can see that DMML is not very sensitive to m in a wide range, and the overall proper results are obtained using embedding with the size in the range of [128, 512] for most datasets. Besides, the performance of DMML with large embedding size (e,g., 1024) also slightly decreases. It may be because too large embedding size inevitably includes more parameters, leading to overfitting problem.

2) IMPACT OF BALANCE FACTOR AND PERTURBATION PARAMETER

We fix K and m at 3 and 512, respectively, in the following to discuss the behaviors of λ and ϵ. The additional parameter λ is introduced when we add our auxiliary loss. We set $\lambda = \{1, 10, 100, 1000\}$ to analyze the performance of DMML with $\epsilon = 0.01$. We show these results in Table 2. It is found that the optimal λ is in the range of [1, 100]. Besides, when λ is too large (e.g., 1000), the classification performance decreases. In fact, metric learning loss has little impact when λ goes to infinity and the model with a large λ can be considered to be trained using only auxiliary loss. This shows the effectiveness of the metric learning loss in our framework.

Then we demonstrate the influence of ϵ on DMML. To this end, we fix λ to 100 and run our method with

TABLE 5. Classification accuracy (%) of 6 distance-based methods and DMML on 28 UCR datasets.

Dataset	#class	#train	#test	Len	EDNN	DTWNN	SWA	SAD	2DDLCSS	GTM	DMML
50words	50	450	455	270	63.1	69.0	71.9	65.9	**74.9**	71.4	65.1
Adiac	37	390	391	176	61.1	60.4	59.2	56.2	42.5	72.6	**80.1**
Beef	5	30	30	470	66.7	66.7	61.6	50.0	53.3	63.3	**81.7**
CBF	3	30	900	128	85.2	99.7	98.7	95.6	98.8	95.9	**99.8**
ChlorineConcentration	3	467	3840	166	65.0	64.8	62.6	56.1	56.1	71.9	**72.3**
Coffee	2	28	28	286	**100.0**	**100.0**	73.0	81.5	89.3	85.7	**100.0**
DiatomsizeReduction	4	16	306	345	93.5	96.7	97.2	**98.4**	88.2	93.5	95.7
ECG200	2	100	100	96	**88.0**	77.0	83.0	74.4	87.0	80.0	82.3
ECGFiveDays	2	23	861	136	79.7	76.8	71.0	73.5	94.4	98.8	**99.8**
FaceFour	4	24	88	350	78.4	83.0	86.6	75.0	81.8	**96.6**	94.3
FacesUCR	14	200	2050	131	76.9	90.5	**97.0**	68.5	91.5	91.5	90.0
Fish	7	175	175	463	78.3	82.3	82.9	85.0	94.3	93.7	**95.7**
GunPoint	2	50	150	150	91.3	90.7	93.4	99.3	96.7	98.7	**99.7**
ItalyPowerDemand	2	67	1029	24	95.5	95.0	91.8	76.7	93.3	92.1	**96.0**
Lighting2	2	60	61	637	75.4	**86.9**	84.0	72.8	82.0	75.4	73.8
Lighting7	7	70	73	319	57.5	72.6	**91.0**	83.3	54.8	42.5	79.5
MALLAT	8	55	2345	1024	91.4	93.4	72.1	44.3	94.0	92.6	**97.4**
MedicalImages	10	381	760	99	68.4	73.7	65.2	56.6	65.5	73.3	**74.0**
MoteStrain	2	20	1252	84	87.9	83.5	92.7	89.7	84.8	89.6	**94.1**
OliveOil	4	30	30	570	86.7	83.3	**90.3**	79.3	16.7	70.0	73.3
SonyAIBORobotSurface	2	20	601	70	69.5	72.5	79.5	80.5	86.7	82.0	**97.6**
SonyAIBORobotSurfaceII	2	27	953	65	85.9	83.1	71.9	67.8	85.7	91.0	**96.4**
SwedishLeaf	15	500	625	128	78.9	79.2	86.0	74.6	89.4	86.2	**95.1**
Syntheticcontrol	6	300	300	60	88.0	99.3	94.0	85.0	94.0	87.7	**99.7**
Trace	4	100	100	275	76.0	**100.0**	89.2	**100.0**	96.0	99.0	**100.0**
TwoLeadECG	2	23	1139	82	74.7	90.4	85.1	98.3	93.1	99.6	**99.9**
TwoPatterns	4	1000	4000	128	91.0	**100.0**	**100.0**	94.8	99.9	84.6	98.8
yoga	2	300	3000	426	83.0	83.6	57.0	87.0	**87.7**	86.9	85.5
Winning times					2	4	4	2	2	1	**18**
Average rank					4.714	3.750	4.143	5.107	3.893	3.750	**2.179**
Average accuracy					79.9	84.1	81.7	77.5	81.2	84.5	**89.9**
Rank difference					2.536	1.571	1.964	2.929	1.714	1.571	-
Wilcoxon test p-value					0.000	0.002	0.000	0.000	0.001	0.001	-

$\epsilon = \{0.0001, 0.001, 0.01, 0.1, 1\}$, respectively. Table 1 illustrates the classification performance with the increasing of ϵ. The best classification performance of DMML is obtained when ϵ is in the range of [0.1, 0.001] for most datasets. Besides, the performance of the model dramatically decreases when $\epsilon = 1$. The reason is that the synthetic perturbations are too large and the synthetic series are too far from the original series. Thus, the synthetic series bring a lot of interference information that disturbs the training and affects the learning of distance metrics.

D. PERFORMANCE COMPARISON

In this section, we compare DMML with four representative types of techniques: (1) distance-based methods, (2) shapelet-based methods, (3) feature-based methods, and (4) deep learning-based methods. As comparison data, we use the reported error rate from the available literature. We report the average performance of DMML over 10 replicates. In addition, we report the rank difference and the Wilcoxon test p-value between DMML and baseline methods. Rank difference measures the gaps between the average ranks of DMML and baseline methods. Wilcoxon test is a non-parametric test which is used to make statistical comparison.

We first compare DMML with 6 distance-based methods, i.e., Euclidean distance based 1-nearest neighbor (EDNN), dynamic time warping based 1-nearest neighbor (DTWNN), sequence weighted alignment (SWA) [54], spatial assembling distance (SAD) [55], longest common subsequence using the first and second derivatives (2DDLCSS) [6], and geometric template matching (GTM) [7]. The classification results of EDNN and DTWNN are reported in [51], and the results of SWA and SAD are reported in [56]. For 2DDLCSS and GTM, the results are reported in [6] and [7], respectively. Table 5 shows the experimental results of DMML and 6 distance-based methods. On 28 datasets, DMML achieves the best accuracy for 18 datasets. There is a is a significant difference between DMML and the distance-based methods according to the rank difference and Wilcoxon test p-value. Besides, the average rank and average accuracy of DMML are 2.179 and 89.9%, respectively. GTM ranks the second with an average rank of 3.750, and its winning times and average accuracy are 1 and 84.5%, respectively, which are lower than DMML. It suggests that our proposed DMML model that learns a data-adaptive distance metric yields better performance compared with other distance-based methods.

Then we compare DMML with 5 shapelet-based time series classification methods, i.e., naive shapelet decision tree

TABLE 6. Classification accuracy (%) of 5 shapelet-based methods and DMML on 15 UCR datasets.

Dataset	#class	#train	#test	Len	NDS	FSH	LTS	SLAS	ELIS	DMML
Beef	5	30	30	470	50.0	53.3	73.3	73.3	63.3	**81.7**
BeetleFly	2	20	20	512	75.0	75.0	75.0	**95.0**	85.0	89.0
BirdChicken	2	20	20	512	85.0	85.0	80.0	**100.0**	90.0	**100.0**
Coffee	2	28	28	286	96.4	92.9	92.9	96.4	96.4	**100.0**
DiatomSizeReduction	4	16	306	345	72.2	68.9	92.5	86.6	89.9	**95.7**
ECGFiveDays	2	23	861	136	77.5	99.7	66.3	97.3	**100.0**	99.8
FaceFour	4	24	88	350	84.1	90.9	**95.5**	93.2	**95.5**	94.3
GunPoint	2	50	150	150	89.3	99.3	98.7	98.7	99.3	**99.7**
ItalyPowerDemand	2	67	1029	24	89.2	92.1	95.6	74.1	**97.6**	96.0
Lighting7	7	70	73	319	49.3	60.3	75.3	68.5	**80.8**	79.5
MoteStrain	2	20	1252	84	82.5	78.5	86.6	82.8	89.8	**94.1**
SonyAIBORobotSurface	2	20	601	70	84.5	69.9	77.7	64.4	87.9	**97.6**
Symbols	6	25	995	398	78.2	93.2	90.1	88.7	78.3	**98.1**
Trace	4	100	100	275	98.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
TwoLeadECG	2	23	1139	82	58.1	92.2	99.7	94.8	99.8	**99.9**
Winning times					0	1	2	3	5	**10**
Average rank					5.067	4.133	3.333	3.400	2.200	**1.400**
Average accuracy					78.0	83.4	86.6	87.6	90.2	**95.0**
Rank difference					3.667	2.733	1.933	2.000	0.800	-
Wilcoxon test p-value					0.000	0.000	0.000	0.002	0.056	-

TABLE 7. Classification accuracy (%) of 5 feature-based methods and DMML on 26 UCR datasets.

Dataset	#class	#train	#test	Len	BOSS	DTWF	TSF	TSBF	LPS	DMML
50words	50	450	455	270	70.2	74.8	72.8	74.4	**77.6**	65.1
Adiac	37	390	391	176	74.9	60.5	70.7	72.7	76.5	**80.1**
Beef	5	30	30	470	61.5	54.6	64.8	55.4	52.0	**81.7**
BirdChicken	2	20	20	512	98.4	86.5	83.9	90.2	85.4	**100.0**
Car	4	60	60	577	85.5	85.1	75.8	79.5	83.6	**89.0**
CBF	3	30	900	128	**99.8**	97.9	95.8	97.7	98.4	**99.8**
ChlorineConcentration	3	467	3840	166	66.0	65.8	71.9	68.3	64.2	**72.3**
Coffee	2	28	28	286	98.9	97.3	98.9	98.2	95.0	**100.0**
DiatomSizeReduction	4	16	306	345	93.9	94.2	94.1	89.0	91.5	**95.7**
ECGFiveDays	2	23	861	136	98.3	90.7	92.2	84.9	84.0	**99.8**
FaceFour	4	24	88	350	**99.6**	90.9	89.1	86.2	88.9	94.3
Fish	7	175	175	463	**96.9**	93.1	80.7	91.3	91.2	95.7
GunPoint	2	50	150	150	99.4	96.4	96.2	96.5	97.2	**99.7**
ItalyPowerDemand	2	67	1029	24	86.6	94.8	95.8	92.6	91.4	**96.0**
Lightning7	7	70	73	319	66.6	67.1	72.3	68.0	63.1	**79.5**
MedicalImages	10	381	760	99	71.5	70.1	**75.7**	70.1	71.0	74.0
MoteStrain	2	20	1252	84	84.6	89.1	87.4	88.6	91.7	**94.1**
OliveOil	4	30	30	570	87.0	86.4	88.3	86.4	**89.2**	73.3
SonyAIBORobotSurface1	2	20	601	70	89.7	88.4	84.5	83.9	84.2	**97.6**
SonyAIBORobotSurface2	2	27	953	65	88.8	85.9	85.6	82.5	85.1	**96.4**
SwedishLeaf	15	500	625	128	91.8	88.5	89.2	90.8	92.6	**95.1**
Symbols	6	25	995	398	96.1	93.0	88.8	94.4	96.0	**98.1**
SyntheticControl	6	300	300	60	96.8	98.6	99.0	98.7	97.2	**99.7**
TwoLeadECG	2	23	1139	82	98.5	95.8	84.2	91.0	92.8	**99.9**
TwoPatterns	4	1000	4000	128	99.1	**100.0**	99.1	97.4	96.7	98.8
Yoga	2	300	3000	426	**91.0**	86.3	86.7	83.5	87.4	85.5
Winning times					4	1	1	0	2	**19**
Average rank					2.800	3.880	3.800	4.480	4.240	**1.600**
Average accuracy					88.1	85.8	85.5	85.1	85.5	**90.8**
Rank difference					1.200	2.280	2.200	2.880	2.640	-
Wilcoxon test p-value					0.001	0.000	0.000	0.000	0.000	-

TABLE 8. Classification accuracy (%) of 5 deep learning-based methods and DMML on 24 UCR datasets.

Dataset	#class	#train	#test	Len	FCN	Encoder	NCNN	t-LeNet	TWIESN	DMML
50words	50	450	455	270	62.7	**72.3**	22.0	12.5	49.6	65.1
Adiac	37	391	310	176	**84.4**	48.4	2.2	2.0	41.6	80.1
Beef	5	30	30	470	69.7	64.3	20.0	20.0	53.7	**81.7**
CBF	3	30	900	128	99.4	94.7	33.2	33.2	89.0	**99.8**
Coffee	2	28	28	286	**100.0**	97.9	51.4	53.6	97.1	**100.0**
DiatomSizeReduction	4	16	306	345	31.3	91.3	30.1	30.1	88.0	**95.7**
ECG200	2	100	100	96	88.9	**92.3**	64.0	64.0	84.2	82.3
ECGFiveDays	2	23	861	136	94.0	94.0	61.8	58.4	91.9	**99.8**
FaceFour	4	24	88	350	92.8	81.5	26.8	29.5	85.5	**94.3**
FacesUCR	14	200	2050	131	**94.6**	87.4	15.3	14.3	64.4	90.0
FISH	7	175	175	463	**95.8**	86.6	13.4	12.6	87.5	95.7
GunPoint	2	50	150	150	**100.0**	93.6	51.3	49.3	96.1	99.7
ItalyPowerDemand	2	67	1029	24	96.1	**96.5**	50.0	49.9	88.0	96.0
Lighting2	2	60	61	637	**73.9**	69.2	55.7	54.1	70.3	73.8
Lighting7	7	70	73	319	**82.7**	62.5	31.0	26.0	66.4	79.5
MALLAT	8	55	2345	1024	96.7	87.6	13.5	12.3	59.6	**97.4**
MoteStrain	2	20	1252	84	93.7	84.0	50.8	53.9	78.5	**94.1**
OliveOil	4	30	30	570	72.3	40.0	38.0	38.0	**79.0**	73.3
SonyAIBORobotSurface	2	20	601	70	96.0	74.3	44.3	42.9	63.8	**97.6**
SonyAIBORobotSurfaceII	2	27	953	65	**97.9**	83.9	59.4	61.7	69.7	96.4
synthetic_control	6	300	300	60	98.5	99.6	29.8	16.7	87.4	**99.7**
TwoPatterns	4	1000	4000	128	**100.0**	86.3	50.0	50.0	85.2	98.8
Trace	4	100	100	275	**100.0**	96.0	35.4	24.0	95.9	**100.0**
yoga	2	300	3000	426	83.9	82.0	53.6	53.6	60.7	**85.5**
Winning times					10	3	0	0	1	**12**
Average rank					1.792	2.875	5.167	5.542	3.583	**1.625**
Average accuracy					87.7	81.9	37.6	35.9	76.4	**90.7**
Rank difference					0.167	1.250	3.542	3.917	1.958	-
Wilcoxon test p-value					0.496	0.000	0.000	0.000	0.000	-

(NSD) [20], fast shapelet (FSH) [21], learning time series shapelets (LTS) [25], sequence learning in all-subsequence space (SLAS) [27], and efficient learning interpretable shapelets (ELIS) [24]. The experiments are conducted on 15 time series datasets. Experimental results of all shapelet-based models are reported in [24]. Table 6 presents the results of 5 shapelet-based methods and DMML. On 15 datasets, DMML achieves the best accuracy on 10 datasets. Besides, the average rank and the average accuracy of DMML are 1.400 and 95.0%, respectively, which are better than other shapelet-based methods. There is a significant difference between DMML and the 5 shapelet-based methods at the 0.1 level according to the Wilcoxon test p-value. The results demonstrate that DMML outperforms these shapelet-based models by notable gains.

Subsequently, 5 feature-based methods are compared with DMML: bag of SFA symbols (BOSS) [57], learned pattern similarity (LPS) [58], time series based on a bag-of-features representation (TSBF) [28], time series forest (TSF) [29], and dynamic time warping distances as features (DTWF) [4]. We conduct experiments on 26 datasets. The classification results of feature-based models are reported in [59]. Table 7 shows the classification results of DMML and 5 feature-based methods. DMML provides the best accuracy on 19 of 26 datasets and achieves the highest average rank of 1.600. Besides, DMML outputs an average accuracy of 90.8%, which is slightly higher than BOSS's average accuracy of 88.1%. According to the rank difference and Wilcoxon test p-value, DMML significantly outperforms the 5 feature-based methods. From the experimental results, it is clear that our method is superior to the feature-based algorithms in terms of classification accuracy.

Finally, we compare DMML with 5 deep learning-based time series classification methods: fully convolutional neural network (FCN), Encoder [60], multiscale convolutional neural network (MCNN) [61], time Le-Net (t-LeNet) [62], and time warping invariant echo state setwork (TWIESN) [63]. For the fair comparison, we use the results provided by [64]. Table 8 summaries the experimental results of 5 deep learning-based models and DMML. DMML is the most accurate on 12 of 24 datasets, while FCN outputs the most accurate results on 10 of 24 datasets. DMML has an average rank of 1.625, which is better than FCN's average rank of 1.792. The performance comparison reveals that DMML outperforms state-of-the-art methods in terms of classification accuracy. Although FCN achieves comparable performance (Wilcoxon test p-value $=$ 0.496), it is difficult to extend it to the datasets with unseen classes. DMML captures the semantic distances between time series, which is suitable for the datasets with classes not in training set.

We further observe that the datasets that DMML performs the best have a small number of classes or a small length of series. However, DMML loses its effectiveness on several datasets, such as 50words and Aiac. The common feature of them is that the numbers of classes are larger than other datasets. The numbers of classes in 50words and Aiac are 50 and 37, respectively, which are the largest among all datasets. It means that DMML is not good at dealing with the dataset with a large number of classes. The reason may be that, with the increase of the number of classes, the proportion of negative pairs increases while that of positive pairs decreases, resulting in an imbalance problem for metric learning. Besides, it can be seen that DMML performs poorly for the datasets with long time series, e.g., BettleFly, OliveOil, and Lighting2. The lengths of series in BettleFly, OliveOil, and Lighting2 are 512, 570, and 637, respectively, which are far longer than the average. The reason for this phenomenon may be that the convolutional network is not suitable for capturing the discriminative features from long series.

V. CONCLUSION

In this paper, we present a deep multiple metric learning (DMML) approach for time series classification. DMML learns multiple local distance metrics through a shared convolutional network. The hard negative mining is introduced to enhance the capability of learned metrics to deal with the ambiguous test data samples and encourage the complementarity of metrics. Besides, we introduce an auxiliary loss to guide the model to learn a discriminative metric. The experiments show that our model can outperform the state-of-the-art methods.

We also show that DMML loses its efficiency in some situations. First, DMML is not good at dealing with the datasets with a large number of classes.The reason may be that the number of negative pairs is much more than that of positive pairs if the number of classes is large. For example, if there exists C classes in a dataset with N training samples and each class contain the same number of samples, i.e., N/C, the number of negative pairs is $N^2/2 - N^2/(2C)$ while that of positive pairs is $N^2/(2C) - N/2$. As can be seen, with the increase of C, the number of positive pairs decreases, while the number of negative pairs increases. The number of positive pairs is $(N^2 + N)/2 - N^2/C$ more than the number of negative samples. Since that most of the training pairs are negative pairs when C is large, the model mainly focuses on increasing the distances between samples. However, the correctness of the relationships between the samples from the same class can not be guaranteed, decreasing the discriminative power of the distance metric. In our future work, we will design a hard positive mining algorithm to generate enough positive pairs to address the imbalance problem. Specifically, for each sample, we generate a hard positive sample that is close to it in the input space and far away from it in the embedding space. The sample pair consisting of the sample and the generated hard positive sample is then used as a supplementary training pair so as to increase the number of positive pairs.

Second, DMML performs poorly in face of long series. The reason may be that we use the same size convolution kernels for all datasets. For CNN, the kernel size is essential to capture the informative features from time series [65]. In this paper, we use the same kernel size for all datasets. The kernel sizes are set as 8, 5, and 3, respectively, for the three convolutional blocks in DMML. Compared with the length of time series in BettleFly, OliveOil, and Lighting2 datasets, the kernel sizes are relatively small. We consider that the small kernel sizes are not able to extract informative features on these datasets and therefore yield bad performance. As a future work, we may develop a multiscale version of DMML, which contains multiple convolutional neural networks with different kernel sizes, to solve this problem. Each convolutional neural network captures the feature at a scale. Thus, we can get the features extracted by kernels with different sizes and avoid time-consuming parameter selection.

REFERENCES

[1] J. Wiens, J. V. Guttag, and E. Horvitz, "Patient risk stratification for hospital-associated c. diff as a time-series classification task," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2012, pp. 476–484.
[2] M. Delgado, M. P. Cuellar, and M. C. Pegalajar, "Multiobjective hybrid optimization and training of recurrent neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 381–403, Apr. 2008.
[3] A. Mellit, A. M. Pavan, and M. Benghanem, "Least squares support vector machine for short-term prediction of meteorological time series," *Theor. Appl. Climatol.*, vol. 111, nos. 1–2, pp. 297–307, Jan. 2013.
[4] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining Knowl. Discovery*, vol. 30, no. 2, pp. 283–312, Mar. 2016.
[5] T. Górecki and M. Luczak, "Non-isometric transforms in time series classification using DTW," *Knowl.-Based Syst.*, vol. 61, pp. 98–108, May 2014.
[6] T. Górecki, "Using derivatives in a longest common subsequence dissimilarity measure for time series classification," *Pattern Recognit. Lett.*, vol. 45, pp. 99–105, Aug. 2014.
[7] J. Frank, S. Mannor, J. Pineau, and D. Precup, "Time series analysis using geometric template matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 740–754, Mar. 2013.
[8] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Deep localized metric learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2644–2656, Oct. 2018.
[9] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1363–1374, Jun. 2016.
[10] H. Chen, F. Tang, P. Ti no, A. G. Cohn, and X. Yao, "Model metric co-learning for time series classification," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 3387–3394.
[11] B. Nguyen, C. Morell, and B. De Baets, "Distance metric learning for ordinal classification based on triplet constraints," *Knowl.-Based Syst.*, vol. 142, pp. 17–28, Feb. 2018.
[12] C.-T. Do, A. Douzal-Chouakria, S. Marié, M. Rombaut, and S. Varasteh, "Multi-modal and multi-scale temporal metric learning for a robust time series nearest neighbors classification," *Inf. Sci.*, vols. 418–419, pp. 272–285, Dec. 2017.
[13] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2612–2620.
[14] C. B. Choy, J. Gwak, S. Savarese, and M. K. Chandraker, "Universal correspondence network," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 2406–2414.
[15] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.

[16] C. Hsieh, L. Yang, Y. Cui, T. Lin, S. J. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. WWW*, Perth, WA, Australia, 2017, pp. 193–201.

[17] X. Wang, Y. Hua, E. Kodirov, G. Hu, and N. M. Robertson, "Deep metric learning by online soft mining and class-aware attention," in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 5361–5368.

[18] W. Zheng, Z. Chen, J. Lu, and J. Zhou, "Hardness-aware deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 72–81.

[19] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2780–2789.

[20] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2009, pp. 947–956.

[21] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 668–676.

[22] X. Renard, M. Rifqi, W. Erray, and M. Detyniecki, "Random-shapelet: An algorithm for fast shapelet discovery," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2015, pp. 1–10.

[23] W. Zalewski, F. Silva, A. G. Maletzke, and C. A. Ferrero, "Exploring shapelet transformation for time series classification in decision trees," *Knowl.-Based Syst.*, vol. 112, pp. 80–91, Nov. 2016.

[24] Z. Fang, P. Wang, and W. Wang, "Efficient learning interpretable shapelets for accurate time series classification," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 497–508.

[25] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 392–401.

[26] L. Hou, J. T. Kwok, and J. M. Zurada, "Efficient learning of timeseries shapelets," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 1209–1215.

[27] T. L. Nguyen, S. Gsponer, and G. Ifrim, "Time series classification by sequence learning in all-subsequence space," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 947–958.

[28] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2796–2802, Nov. 2013.

[29] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," *Inf. Sci.*, vol. 239, pp. 142–153, Aug. 2013.

[30] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 3026–3037, Dec. 2014.

[31] S. Lin and G. C. Runger, "GCRNN: Group-constrained convolutional recurrent neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4709–4718, Oct. 2018.

[32] Q. Ma, L. Shen, W. Chen, J. Wang, J. Wei, and Z. Yu, "Functional echo state network for time series classification," *Inf. Sci.*, vol. 373, pp. 1–20, Dec. 2016.

[33] L. Wang, Z. Wang, and S. Liu, "An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm," *Expert Syst. Appl.*, vol. 43, pp. 237–249, Jan. 2016.

[34] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. ICML*, 2007, pp. 209–216.

[35] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[36] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2666–2672.

[37] P. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. ICML*, New York, NY, USA, 2016, pp. 2464–2471.

[38] J. Zhang and L. Zhang, "Efficient stochastic optimization for low-rank distance metric learning," in *Proc. AAAI*, San Francisco, CA, USA, 2017, pp. 933–940.

[39] Z. Feng, R. Jin, and A. Jain, "Large-scale image annotation by efficient and robust kernel metric learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1609–1616.

[40] P. Zhu, R. Qi, Q. Hu, Q. Wang, C. Zhang, and L. Yang, "Beyond similar and dissimilar relations : A kernel regression formulation for metric learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3242–3248.

[41] D.-Y. Yeung and H. Chang, "A Kernel approach for semisupervised metric learning," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 141–149, Jan. 2007.

[42] F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang, "A kernel classification framework for metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 1950–1962, Sep. 2015.

[43] H.-X. Yu, A. Wu, and W.-S. Zheng, "Unsupervised person re-identification by deep asymmetric metric embedding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 956–973, Apr. 2020.

[44] J. Lu, J. Hu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4269–4282, Sep. 2017.

[45] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012.

[46] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *Proc. AAAI*, 2014, pp. 2078–2084.

[47] Z. Gong, H. Chen, B. Yuan, and X. Yao, "Multiobjective learning in the model space for time series classification," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 918–932, Mar. 2019.

[48] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[49] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "Deep metric learning with BIER: Boosting independent embeddings robustly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 276–290, Feb. 2020.

[50] S. Chen, C. Gong, J. Yang, X. Li, Y. Wei, and J. Li, "Adversarial metric learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2021–2027.

[51] H. A. Dau, A. J. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. J. Keogh, "The UCR time series archive," 2018, *arXiv:1810.07758*. [Online]. Available: https://arxiv.org/abs/1810.07758

[52] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.

[53] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and MitigatingMobile sensing heterogeneities for activity recognition," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 127–140.

[54] M. D. Morse and J. M. Patel, "An efficient and accurate method for evaluating time series similarity," in *Proc. ACM SIGMOD Int. Conf. Manage. Data - SIGMOD*, 2007, pp. 569–580.

[55] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. H. Tung, "SpADe: On shape-based pattern detection in streaming time series," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 786–795.

[56] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 275–309, Mar. 2013.

[57] P. Schäfer, "The BOSS is concerned with time series classification in the presence of noise," *Data Mining Knowl. Discovery*, vol. 29, no. 6, pp. 1505–1530, Nov. 2015.

[58] M. G. Baydogan and G. Runger, "Time series representation and similarity based on local autopatterns," *Data Mining Knowl. Discovery*, vol. 30, no. 2, pp. 476–509, Mar. 2016.

[59] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining Knowl. Discovery*, vol. 31, no. 3, pp. 606–660, May 2017.

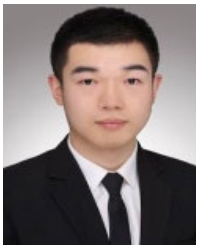
[60] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," in *Proc. CCIA*, Catalonia, Spain, 2018, pp. 120–129.

[61] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," 2016, *arXiv:1603.06995*. [Online]. Available: https://arxiv.org/abs/1603.06995

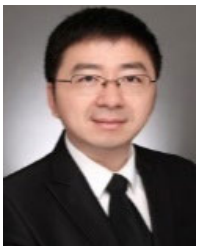
[62] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in *Proc. ECML/PKDD Workshop Adv. Anal. Learn. Temporal Data*, 2016, pp. 1–8.

[63] P. Tanisaro and G. Heidemann, "Time series classification using time warping invariant echo state networks," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 831–836.

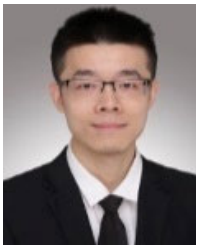
[64] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.
[65] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, "Rethinking 1D-CNN for time series classification: A stronger baseline," 2020, *arXiv:2002.10061*. [Online]. Available: https://arxiv.org/abs/2002.10061



ZHI CHEN received the B.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, China, in 2018. He is currently pursuing the Ph.D. degree with the University of Electronic Science and Technology of China. His current research interests include biological data analysis, machine learning, and data mining.



YONGGUO LIU received the B.S. degree in mechanical engineering from the Sichuan Institute of Light Industry and Chemical Technology, in 1997, the M.S. degree in mechanical engineering from Sichuan University, in 2000, and the Ph.D. degree in computer science from Chongqing University, in 2003. He finished his Postdoctoral Research with Shanghai Jiaotong University, in 2005. He joined the University of Electronic Science and Technology of China, in 2005, where he is currently a Full Professor with the School of Information and Software Engineering. His research interests include medical informatics and data mining.



JIAJING ZHU received the B.S. degree from the School of Information and Software Engineering from the University of Electronic Science and Technology of China, China, in 2015. He is currently pursuing the Ph.D. degree with the University of Electronic Science and Technology of China. His research interests include TCM informatics, machine learning, and pattern recognition.

YUN ZHANG received the B.S. degree from the Software School, in 2014, and the M.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2016. He is currently pursuing the Ph.D. degree with the University of Electronic Science and Technology of China. His research interests include complex network analysis, natural language processing, and TCM informatics.

QIAOQIN LI received the M.S. and Ph.D. degrees in computer application from the University of Electronic Science and Technology of China, China, in 2000 and 2010, respectively. She is currently an Associate Professor with the School of Software and Information Engineering, University of Electronic Science and Technology of China. Her current research interests include artificial neural networks, machine learning, and the Internet of Things.

RONGJIANG JIN received the B.S. degree in Chinese medicine and the M.S. degree in acupuncture from the Chengdu College of Traditional Chinese Medicine, in 1984 and 1994, respectively, and the Ph.D. degree in acupuncture moxibustion and tuina from the Chengdu University of Traditional Chinese Medicine, in 2005. He joined the Chengdu University of Traditional Chinese Medicine, in 1984, where he is currently the Dean and a Professor with the School of Health Preservation and Rehabilitation. His research interests include acupuncture, moxibustion, and traditional Chinese rehabilitation.

XIA HE received the B.S. degree in Chinese medicine from the Chengdu University of Traditional Chinese Medicine, in 2003. She is currently the Vice President of the Sichuan Bayi Rehabilitation Center and an Adjunct Professor with the Chengdu University of Traditional Chinese Medicine. Her research interests include integrated traditional Chinese and western medicine clinical rehabilitation for neurological disorders and senile diseases.