

基于原型动态对齐的时间序列分类方法

陈 涛

温州大学计算机与人工智能学院, 浙江 温州

收稿日期: 2026年5月20日; 录用日期: 2026年7月6日; 发布日期: 2026年7月17日

摘 要

随着传感器技术、智能医疗和工业监测的发展, 时间序列数据在脑电信号分析、设备状态识别和行为检测等领域中的应用日益广泛, 其动态变化和时间错位特征给分类任务带来了较大挑战。传统深度学习方法通常依赖大量标注样本, 在低资源场景下容易出现过拟合; 而动态时间规整方法虽然能够处理时间错位问题, 但难以通过训练过程持续优化模型表示。因此, 本文提出一种基于原型动态对齐网络的时间序列分类方法。该方法首先利用特征编码模块提取时间序列的潜在表示, 然后构建可学习类别原型以刻画不同类别的典型时序结构, 并引入软动态时间对齐机制计算样本与类别原型之间的匹配关系。实验评估在多个公开UCR时间序列数据集上进行, 结果表明, 所提方法在低资源训练条件下具有较好的分类稳定性, 并在训练样本较充分时仍保持一定竞争力, 验证了模型的有效性。

关键词

时间序列分类, 动态时间规整, 原型学习, 小样本学习, 神经网络

Time Series Classification Based on Prototype Dynamic Alignment

Tao Chen

College of Computer and Artificial Intelligence, Wenzhou University, Wenzhou Zhejiang

Received: May 20, 2026; accepted: July 6, 2026; published: July 17, 2026

Abstract

With the development of sensor technology, intelligent healthcare, and industrial monitoring, time series data have been widely used in electroencephalogram analysis, equipment state recognition, behavior detection, and other fields. Their dynamic variations and temporal misalignment bring considerable challenges to classification tasks. Traditional deep learning methods usually rely on a large number of labeled samples and are prone to overfitting in low-resource scenarios. Although

dynamic time warping can handle temporal misalignment, it is difficult to continuously optimize model representations through training. Therefore, this paper proposes a time series classification method based on a prototype dynamic alignment network. The proposed method first uses a feature encoding module to extract latent representations of time series, then constructs learnable class prototypes to describe typical temporal structures of different categories, and introduces a soft dynamic time alignment mechanism to measure the matching relationship between samples and class prototypes. Experiments are conducted on multiple public UCR time series datasets. The results show that the proposed method achieves stable classification performance under low-resource training conditions and remains competitive when sufficient training samples are available, which verifies the effectiveness of the model.

Keywords

Time Series Classification, Dynamic Time Warping, Prototype Learning, Few-Shot Learning, Neural Network

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

时间序列分类是机器学习领域中的重要研究任务，其目标是根据时间序列中蕴含的动态变化模式为样本分配相应的类别标签。随着传感器技术、智能医疗和工业监测系统的发展，时间序列数据广泛存在于脑电信号分析、心电信号识别、人体行为识别、设备故障诊断和金融趋势分析等实际场景中[1][2]。与普通静态数据不同，时间序列不仅包含数值特征，还包含明显的时间依赖关系和局部动态模式。因此，如何从序列数据中提取具有判别性的时序特征，并实现准确分类，已经成为当前智能数据分析中的重要问题[3]。

近年来，深度学习方法在时间序列分类任务中取得了较好效果[4]。卷积神经网络能够从局部时间窗口中提取模式特征，循环神经网络能够建模序列前后时刻之间的依赖关系[5]，InceptionTime、ROCKET等方法也在公开时间序列分类任务中表现出较强的分类能力[6][7]。这类方法通常依靠端到端训练方式自动学习特征表示，减少了人工特征设计的依赖，并在样本数量较充足的情况下具有较高的分类精度[8]。然而，深度学习模型也存在明显不足。一方面，这类模型通常需要较多标注样本才能获得稳定的泛化能力；另一方面，在低资源或冷启动场景中，训练样本数量有限，模型容易过拟合，导致分类性能不稳定[9]。对于医学时间序列、脑电信号和部分稀有事件识别任务而言，数据采集和标注成本较高，标注样本不足的问题尤为突出[10][11]。因此，仅依赖传统深度神经网络并不能很好地满足低资源时间序列分类需求。

与深度学习方法不同，动态时间规整(Dynamic Time Warping, DTW)是一种经典的时间序列相似性度量方法[12]。DTW 可以通过动态规划寻找两条序列之间的最优对齐路径，从而缓解时间偏移、局部拉伸和模式错位对相似性计算的影响[13]。由于时间序列在实际采集过程中常常存在速度变化、局部延迟和形态差异，DTW 在序列匹配和分类任务中具有较强的适用性[14]。基于 DTW 的最近邻分类方法不需要复杂训练过程，即使在样本数量较少的情况下也能直接进行分类，因此在低资源场景中具有一定优势[15]。此外，DTW 的对齐路径能够直观展示序列之间的匹配关系，相比部分黑箱神经网络模型具有更好的可解释性。

然而,传统DTW方法也存在一定局限。首先,DTW本质上是一种非参数方法,主要依靠样本之间的距离计算完成分类,难以通过训练过程主动学习更适合任务的数据表示。其次,DTW通常直接作用于原始序列或固定特征序列,容易受到噪声、冗余片段和局部异常值的影响。再次,当训练样本逐渐增加时,传统DTW方法难以像神经网络一样通过参数优化持续提升模型性能[16]。对于实际时间序列分类任务而言,一个理想模型应同时具备两方面能力:一方面能够继承DTW对时间错位的适应能力,在低资源场景下保持稳定;另一方面又能够像神经网络一样进行端到端训练,在有更多标注数据时进一步优化模型表示。

为了解决上述问题,本文提出一种基于原型动态对齐的时间序列分类方法。该方法将原型学习、神经网络特征表示和软动态时间对齐机制结合起来,用于提升低资源条件下的时间序列分类性能。原型学习能够通过类别原型表示不同类别的典型模式,从而减少模型对单个训练样本的依赖[17];软动态时间规整方法则通过可微形式对DTW进行平滑化处理,使动态对齐过程能够参与神经网络训练[18]。具体而言,模型首先通过特征编码模块对输入时间序列进行表示学习,将原始序列映射到更稳定的隐空间中;然后为每个类别构建可学习原型,用于刻画该类别的典型时序结构;在此基础上,采用软动态时间对齐机制计算样本序列与类别原型之间的匹配关系,并根据对齐距离完成分类决策。与传统DTW方法相比,本文方法不再完全依赖原始样本作为匹配模板,而是通过可学习原型提取类别代表模式;与普通深度学习模型相比,本文方法显式引入动态对齐机制,能够更好地处理序列中的局部时间错位问题。

本文的主要贡献如下。

- 1) 提出一种基于原型动态对齐的时间序列分类方法。该方法将神经网络表示学习与动态时间对齐思想相结合,使模型既具有一定可训练性,又能够保留对时间错位的建模能力。
- 2) 构建可学习类别原型表示机制。通过类别原型描述不同类别的典型时序模式,减少模型对单个训练样本的依赖,提高低资源场景下的分类稳定性。
- 3) 引入软动态时间对齐策略。通过可微形式的动态对齐计算样本与原型之间的匹配关系,使模型能够在训练过程中联合优化特征表示和类别原型。
- 4) 在公开时间序列分类数据集上开展实验验证。实验结果表明,所提方法在低资源训练条件下具有较好的适应能力,并在数据较充分的情况下仍保持一定竞争力。

2. 方法

在本节中,我们提出一种基于原型动态对齐网络的时间序列分类方法,记为PDAN。该方法主要面向低资源条件下的时间序列分类任务,旨在解决深度学习模型在样本数量不足时容易过拟合、传统动态时间规整方法难以端到端训练等问题。在2.1节中,我们介绍PDAN方法的整体模型框架;在2.2节中,介绍时间序列特征编码模块;在2.3节中,介绍可学习类别原型构建方法;在2.4节中,介绍软动态时间对齐机制;在2.5节中,介绍模型的分类决策与训练策略。

2.1. 模型整体结构

PDAN方法的整体模型框架如图1所示。PDAN主要包含三个核心部分。第一部分是时间序列特征编码模块。第二部分是动态时间对齐模块。第三部分是分类决策模块。此外,模型中还包含原型注意力机制,用于增强原型表示能力。模型的输入是一组时间序列样本,每个样本可以表示为:

$$X = \{x_1, x_2, \dots, x_T\}, x_T \in \mathbb{R}^C \quad (1)$$

其中, T 表示时间序列长度, C 表示变量维度。当处理单变量时间序列时, $C=1$;当处理多变量时间序列时, C 表示通道数或变量数。模型的目标是根据输入序列 X 预测其对应的类别标签 y 。

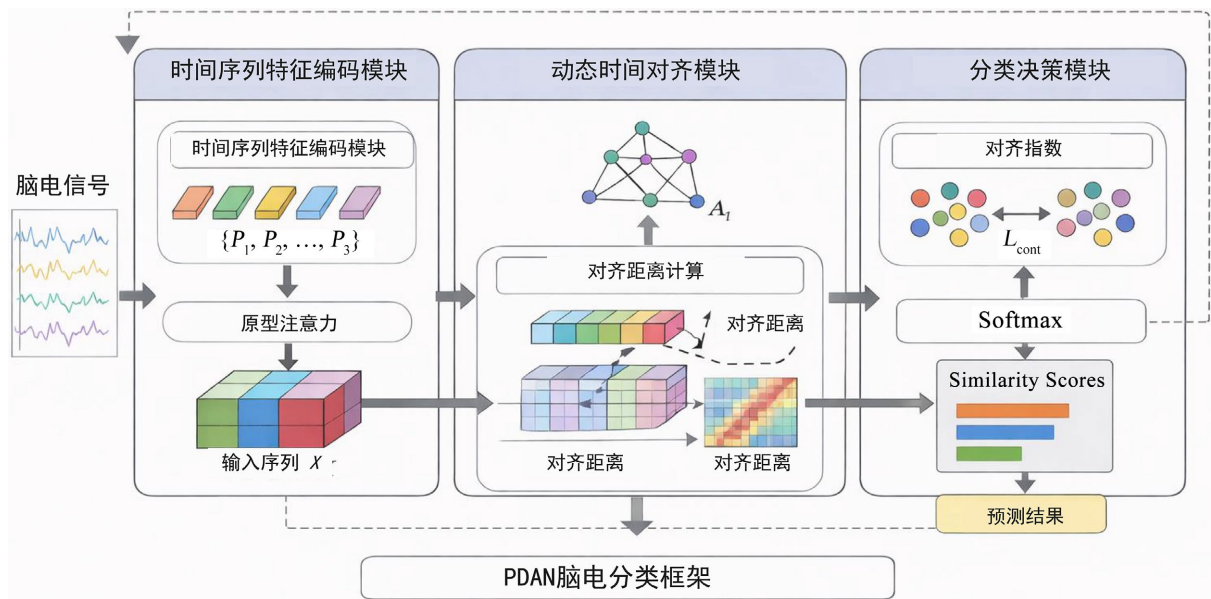


Figure 1. Framework of PDAN for time series classification

图 1. PDAN 时间序列分类框架

首先，输入时间序列经过特征编码模块，得到隐空间中的序列表示。与直接使用原始序列进行距离计算不同，特征编码模块可以对原始数据中的噪声和冗余信息进行一定过滤，从而提取更加稳定的时序特征。其次，模型为每个类别构建一个或多个可学习原型序列。原型序列用于表示该类别中具有代表性的时间结构模式，相当于类别层面的参考模板。然后，模型采用软动态时间对齐机制计算输入序列与各类别原型之间的对齐距离。该机制能够在保留动态时间规整处理时间错位优势的同时，使对齐过程具备可微性，从而支持神经网络的端到端训练。最后，模型根据输入序列与不同类别原型之间的对齐距离生成分类概率，并输出最终预测结果。

从整体流程来看，PDAN 并不是简单地将动态时间规整方法直接用于原始样本匹配，而是通过神经网络表示学习和可学习原型机制，将传统基于距离的时间序列分类思想扩展为可训练模型。该设计使模型既能够利用动态时间对齐方法对局部时间偏移的适应能力，又能够通过参数优化提升类别表示质量。

PDAN 的主要处理流程如下：

- 1) 输入原始时间序列样本，并通过特征编码模块获得隐空间表示；
- 2) 根据类别信息构建类别原型，并在训练过程中不断优化原型参数；
- 3) 利用软动态时间对齐机制计算输入序列与不同类别原型之间的对齐距离；
- 4) 将对齐距离转换为分类概率，并通过损失函数进行模型训练。

2.2. 时间序列特征编码

时间序列数据通常具有非平稳性、局部波动性和噪声干扰等特点。如果直接在原始序列上进行距离计算，模型容易受到局部异常值和无关片段的影响。因此，本文首先采用特征编码模块对输入序列进行表示学习，将原始时间序列映射到更适合分类任务的隐空间。

本文采用三层一维卷积结构作为特征编码器。每层卷积后依次连接批量归一化层和 ReLU 激活函数，用于提取不同尺度下的局部时序特征。其中，卷积核大小可设置为 8、5 和 3，通道数分别设置为 128、256 和 128。该结构能够在较低模型复杂度下提取时间序列中的局部变化模式，并通过非线性映射获得更具判别性的隐空间表示。与直接在原始序列上计算 DTW 距离相比，经过编码后的特征能够减少噪声和

异常片段对齐结果的影响。

设输入时间序列为 X ，特征编码模块记为 $f_\theta(\cdot)$ ，其中 θ 表示可学习参数。编码后的序列表示为：

$$Z = f_\theta(X) \quad (2)$$

其中， $Z = \{z_1, z_2, \dots, z_T\}$ ， $z \in R^d$ ， d 表示隐空间特征维度。编码器可以由一维卷积网络、循环神经网络或其他序列特征提取结构构成。通过编码器处理后，模型能够在保留时间结构的基础上提取更加抽象的特征表示。

在本文方法中，特征编码模块具有两个作用。第一，它能够对原始时间序列进行特征变换，使模型不再依赖原始数值空间中的简单距离计算。第二，它可以与后续原型学习和动态对齐模块共同训练，使编码后的特征更加适合类别区分。对于低资源时间序列分类任务而言，稳定的特征表示能够减少模型对少量训练样本中偶然噪声的依赖，从而提高分类稳定性。

2.3. 可学习类别原型

传统的最近邻动态时间规整方法通常直接将训练样本作为参考序列，并通过计算测试样本与训练样本之间的 DTW 距离完成分类。这种方式虽然在小样本情况下具有一定优势，但也存在明显不足。当训练样本中包含噪声或异常片段时，模型会直接受到参考样本质量的影响；同时，随着训练样本数量增加，逐一匹配所有样本也会带来较大的计算开销。

为此，本文引入可学习类别原型机制。类别原型用于表示每个类别的典型时序模式。与直接使用某个训练样本作为模板不同，PDAN 将原型序列视为可学习参数，并在训练过程中不断更新，使其逐渐逼近类别中心结构。设数据集中共有 K 个类别，第 k 类的原型表示为：

$$P_k = \{p_{k,1}, p_{k,2}, \dots, p_{k,L}\} \quad (3)$$

其中， L 表示原型序列长度， $p_{k,j} \in R^d$ 表示第 k 类原型在第 j 个时间位置上的特征向量。

在模型初始化阶段，可以采用类别样本平均策略构建初始原型。设第 k 类训练样本集合为 S_k ，则其初始原型可表示为：

$$P_k^{(0)} = \frac{1}{|S_k|} \sum_{X_i \in S_k} f_\theta(X_i) \quad (4)$$

其中， $|S_k|$ 表示第 k 类样本数量。该初始化方式能够减少随机初始化带来的不稳定性，使模型在训练初期获得较好的类别参考表示。在训练过程中，类别原型与特征编码网络参数一起通过反向传播进行更新。这样，模型不仅能够学习输入序列的特征表示，还能够学习每个类别更具代表性的原型模式。PDAN 不直接把训练样本作为参考序列，而是学习可优化的原型序列，以提高小样本条件下的稳定性和分类性能。

为了进一步增强类别原型的判别能力，可以引入原型约束，使同类别样本与对应原型之间的距离尽可能减小，不同类别原型之间保持一定差异。这样可以避免不同类别原型过于接近，从而提升分类边界的清晰度。

2.4. 软动态时间对齐机制

时间序列分类中的一个重要问题是局部时间错位。即使两个样本属于同一类别，其关键模式也可能出现在不同时间位置。如果直接采用欧氏距离等逐点匹配方式，模型难以处理这种局部偏移问题。动态时间规整方法能够通过动态规划寻找两条序列之间的最优匹配路径，因此适合用于处理时间序列中的局部拉伸、压缩和错位现象。

给定输入序列表示 $Z = \{z_1, z_2, \dots, z_T\}$ 和第 k 类原型序列 $P_k = \{p_{k,1}, p_{k,2}, \dots, p_{k,L}\}$ ，首先计算二者之间的局部距离矩阵：

$$D_{t,j}^{(k)} = \|z_t - p_{k,j}\|_2^2 \quad (5)$$

其中, $D_{t,j}^{(k)}$ 表示输入序列第 t 个时间点与第 k 类原型第 j 个时间点之间的局部距离。

传统 DTW 通过动态规划计算累积距离, 其递推公式为:

$$R_{t,j} = D_{t,j} + \min \{ R_{t-1,j}, R_{t,j-1}, R_{t-1,j-1} \} \quad (6)$$

其中, $R_{t,j}$ 表示从起点到位置 (t,j) 的最小累积距离。该公式能够有效获得最优对齐路径, 但其中的最小值操作不连续可导, 不利于神经网络中的梯度传播。

为了解决这一问题,本文采用软动态时间对齐策略。该策略使用平滑化的 soft-min 函数替代传统 DTW 中的硬最小值操作,使动态对齐过程能够参与端到端训练。 soft-min 函数定义为:

$$\text{softmax}_{\gamma}(a_1, a_2, \dots, a_n) = -\gamma \log \sum_{i=1}^n \exp\left(-\frac{a_i}{\gamma}\right) \quad (7)$$

其中, γ 为平滑参数, 用于控制 **soft-min** 对真实最小值的逼近程度。当 γ 趋近于 0 时, **soft-min** 函数逐渐接近传统最小值函数; 当 γ 较大时, 函数更加平滑。

引入 soft-min 后, 软动态时间对齐的递推公式为:

$$R_{t,i}^\gamma = D_{t,i} + \text{softmin}_\gamma = (R_{t-1,i}^\gamma, R_{t,i-1}^\gamma, R_{t-1,i-1}^\gamma) \quad (8)$$

最终，输入序列与第 k 类原型之间的对齐距离记为：

$$d_k = R_{T-L}^\gamma \quad (9)$$

通过上述方式，模型可以在隐空间中计算样本与类别原型之间的柔性匹配距离。该机制既保留了 DTW 处理时间错位的优势，又解决了传统 DTW 不适合直接用于神经网络训练的问题。Soft-DTW 使用平滑化最小值函数替代传统 DTW 的硬最小值操作，使动态规划过程变为可微函数，从而能够稳定传播梯度。神经网络中的计算流程如图 2 所示。

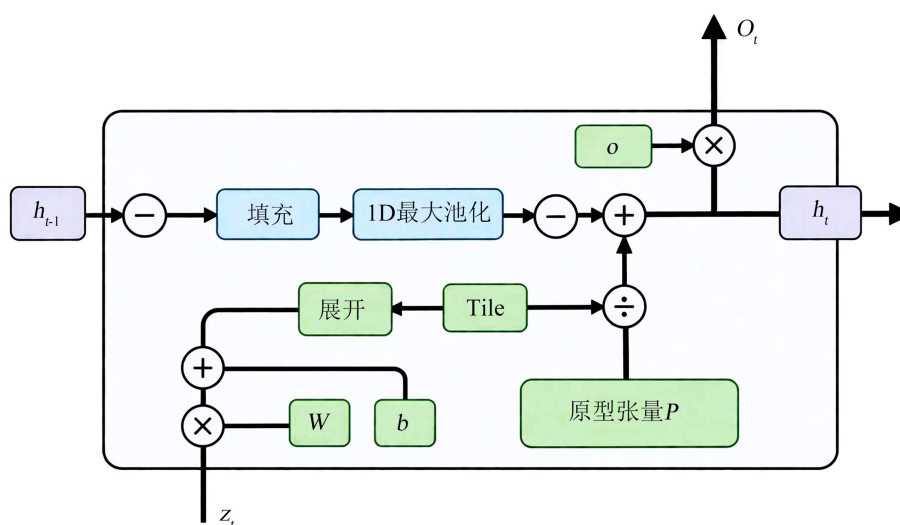


Figure 2. Computation flow in the neural model
图 2. 神经模型中的计算流程

2.5. 分类决策与模型训练

在得到输入序列与各类别原型之间的对齐距离后，模型需要将这些距离转换为类别概率。设输入序列与第 k 类原型之间的对齐距离为 d_k ，距离越小表示输入样本与该类别原型越相似。因此，本文采用负距离作为分类得分，并通过 Softmax 函数得到类别概率：

$$\hat{y}_k = \frac{\exp(-d_k)}{\sum_{c=1}^K \exp(-d_c)} \quad (10)$$

其中， K 表示类别数量， $\hat{y}_k$ 表示模型预测样本属于第 k 类的概率。最终预测类别为：

$$\hat{y} = \arg \max_k \hat{y}_k \quad (11)$$

模型训练时采用交叉熵损失函数作为主要分类损失：

$$L_{cls} = -\sum_{k=1}^K y_k \log \hat{y}_k \quad (12)$$

其中， y_k 表示真实标签的独热编码。为了提高类别原型的判别性，还可以加入原型约束损失。该约束一方面鼓励样本靠近其真实类别原型，另一方面鼓励不同类别原型之间保持一定距离。模型总体损失可表示为：

$$L = L_{cls} + \lambda L_{proto} \quad (13)$$

其中， L_{proto} 表示原型约束损失， λ 为权重系数，用于平衡分类损失和原型约束损失。

完整训练流程如下：

- 1) 输入训练时间序列样本，并通过特征编码模块得到隐空间表示；
- 2) 根据类别标签初始化类别原型，并将其设置为可学习参数；
- 3) 计算输入序列与各类别原型之间的局部距离矩阵；
- 4) 通过软动态时间对齐机制得到样本与每个类别原型之间的对齐距离；
- 5) 将对齐距离转换为分类概率，并计算分类损失；
- 6) 通过反向传播更新特征编码网络参数和类别原型参数；
- 7) 重复训练过程，直到模型收敛或达到最大训练轮数。

通过上述训练方式，PDAN 能够在样本数量有限时利用类别原型增强分类稳定性，同时通过软动态时间对齐机制处理序列中的局部时间偏移。与传统 DTW 最近邻方法相比，PDAN 具有可训练性；与普通深度学习模型相比，PDAN 显式建模了序列与类别原型之间的动态匹配关系，因此更适合低资源时间序列分类任务。

3. 结果

在本节中，我们通过公开时间序列分类数据集对所提出的 PDAN 方法进行实验验证。3.1 节介绍实验数据集、对比方法和实验设置；3.2 节给出 PDAN 与多种基线方法的整体分类性能对比；3.3 节进一步分析模型在低资源训练条件和完整训练条件下的表现。

3.1. 数据集与实验设置

为了验证 PDAN 方法在时间序列分类任务中的有效性，本文选取 UCR 时间序列分类数据集作为实验数据来源。UCR 数据集是时间序列分类领域常用的公开基准数据集，涵盖医学信号、传感器数据、图

像轮廓序列、运动轨迹等多种类型的时间序列任务，能够较全面地评估分类模型的泛化能力和适应能力。

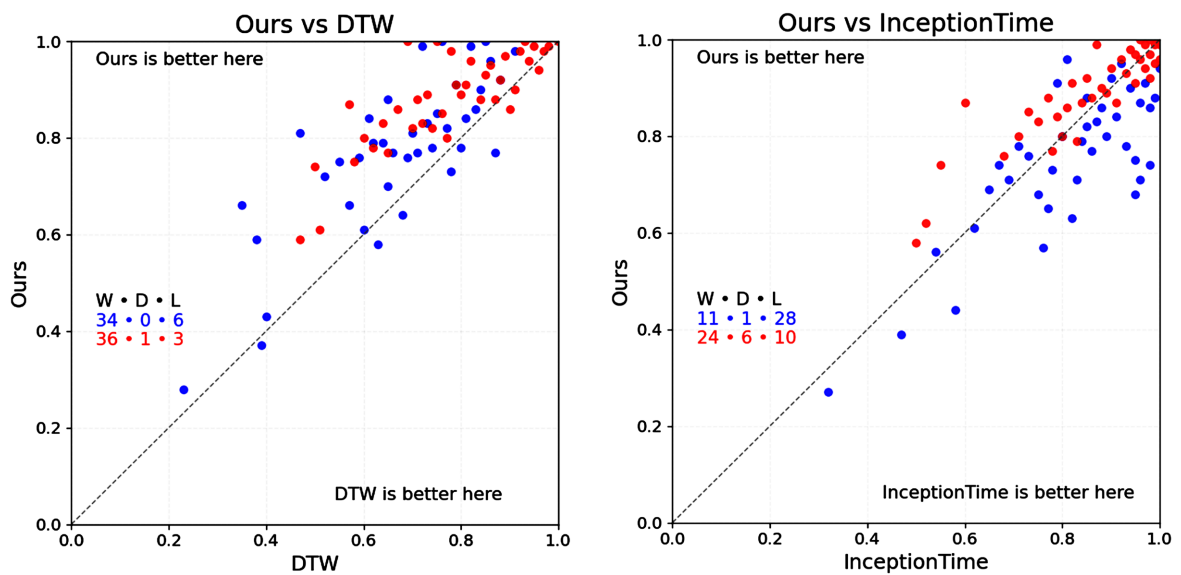
为了避免实验结果受到数据集选择偏差的影响，本文在整体实验中首先使用 UCR 数据集中的 80 个基准数据集进行综合评估。在进一步分析动态对齐类方法性能时，选取其中 40 个距离型时间序列数据集作为重点分析对象。这里的“距离型数据集”主要指在传统距离度量方法，尤其是 NN-DTW 方法下具有较强竞争力的数据集。这类数据集通常具有明显的局部时间偏移、形态伸缩或序列对齐需求，适合用于评估动态时间对齐机制的有效性。本文并未仅依据 PDAN 的实验结果进行筛选，而是根据数据集的时序结构特点以及距离度量方法在该类任务中的适用性进行划分，以保证对比分析的合理性。

同时为了验证模型在不同训练数据规模下的适应能力，本文设置了低资源训练和完整训练两类实验场景。在低资源实验中，分别随机选取训练集的 1% 和 10% 作为可用训练样本，用于模拟标注样本稀缺或冷启动场景；在完整训练实验中，使用全部训练样本进行模型训练，用于评估模型在数据较充分条件下的学习能力。由于部分 UCR 数据集的训练样本数量较少，难以在各类别中进行均衡采样，因此低资源实验主要选取训练集中每个类别样本数量相对充足的 6 个数据集进行分析，包括 ECG5000、HandOutlines、MiddlePhalanxCorrect、StarLightCurves、Strawberry 和 Wafer。

本文选择多种经典和主流时间序列分类方法作为对比模型，包括最近邻动态时间规整方法(NN-DTW)、长短期记忆网络(LSTM) [5]、InceptionTime [6]、ROCKET [7]、BOSS [19] 和 Elastic Ensemble (EE) [15]。其中，NN-DTW 是传统距离度量方法的代表，能够较好地处理时间序列中的局部错位；LSTM 是典型循环神经网络模型；InceptionTime 和 ROCKET 是近年来表现较好的深度时间序列分类模型；BOSS 和 EE 则代表集成式时间序列分类方法。通过与上述方法进行比较，可以从传统距离方法、循环神经网络、深度学习方法和集成方法等多个角度评价 PDAN 的分类性能。

3.2. 整体分类性能对比

为了直观比较 PDAN 与不同基线方法在公开数据集上的分类效果，本文统计了各模型在 UCR 数据集上的分类准确率，并重点分析 40 个距离型时间序列数据集上的胜负情况。整体对比结果如图 3 所示。图中每个点表示一个数据集，横轴表示基线方法的分类准确率，纵轴表示 PDAN 的分类准确率，红点表示距离型时间序列数据集。若数据点位于对角线上方，说明 PDAN 在该数据集上的准确率高于对应基线方法；若数据点位于对角线下方，则说明基线方法表现更优。



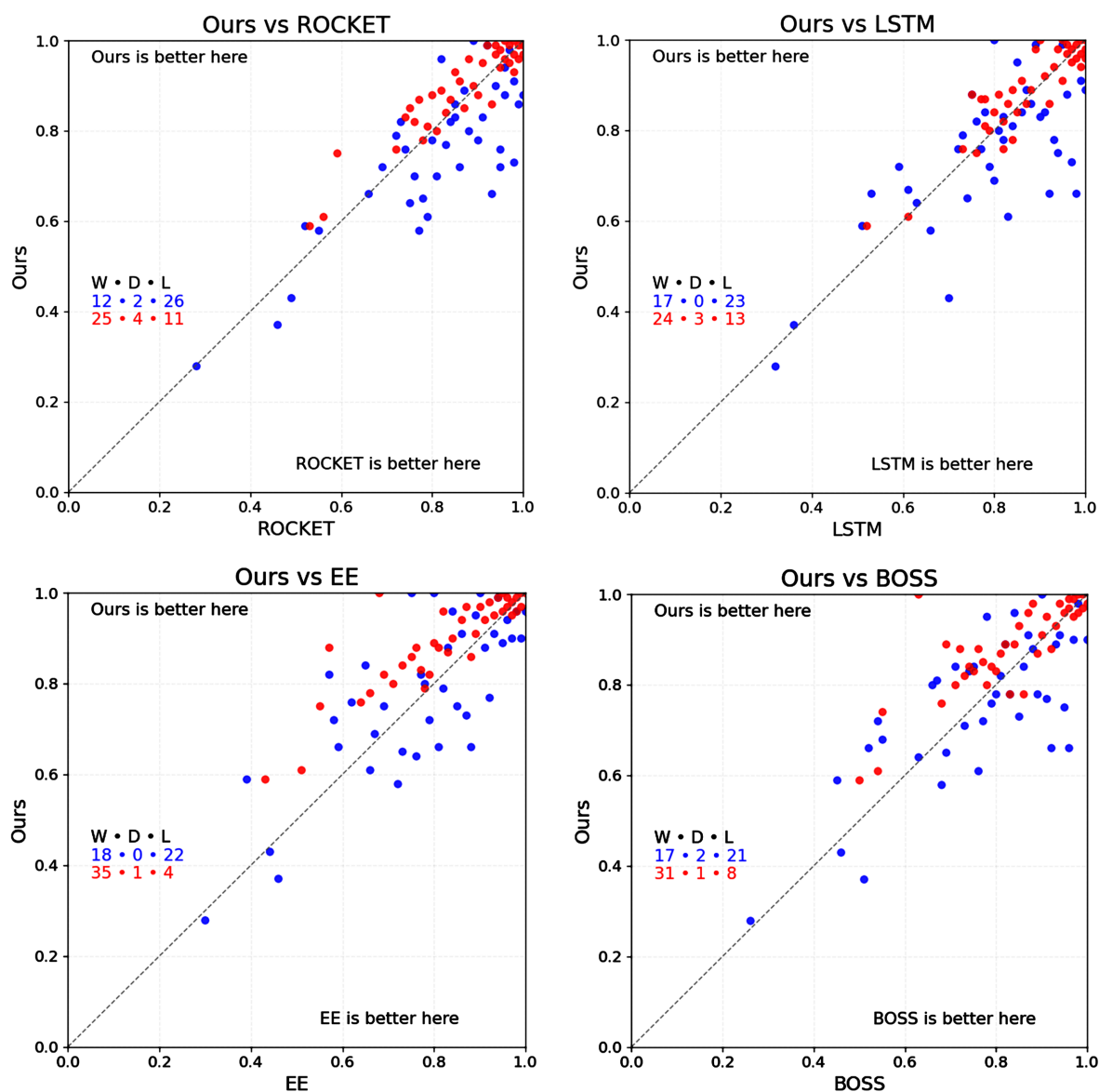


Figure 3. Accuracy comparison between PDAN and different baselines

图 3. PDAN 与不同基线方法的准确率对比

从整体结果可以看出,PDAN 在多个距离型时间序列数据集上均表现出较好的分类性能。与 NN-DTW 相比,PDAN 在 40 个距离型数据集中取得 34 胜、5 平、1 负的结果,表现出明显优势。这说明 PDAN 在保留动态时间对齐思想的同时,通过特征编码和可学习原型机制进一步提升了模型的表示能力。NN-DTW 虽然能够直接处理时间错位问题,但其本质上属于非参数方法,难以根据训练数据主动优化类别表示;而 PDAN 通过神经网络训练过程不断更新特征编码器和类别原型,因此能够在多数距离型数据集上取得更好的分类效果。

与 LSTM 相比,PDAN 在 40 个距离型数据集中取得 26 胜、6 平、8 负的结果。LSTM 能够建模序列前后时刻之间的依赖关系,但其分类效果通常依赖较充分的训练样本。当训练数据规模有限,或者序列中存在局部时间偏移时,LSTM 容易受到样本数量和噪声的影响。相比之下,PDAN 显式引入动态时间对齐机制,可以更直接地处理序列局部错位问题,因此在多数距离型数据集上表现更加稳定。

与深度学习方法相比，PDAN 也具有一定竞争力。相较于 InceptionTime，PDAN 在 40 个距离型数据集中取得 28 胜、4 平、8 负的结果；相较于 ROCKET，PDAN 取得 25 胜、5 平、10 负的结果。InceptionTime 和 ROCKET 在时间序列分类任务中具有较强性能，尤其擅长通过卷积结构或随机卷积核提取局部模式。然而，这类方法通常更侧重于局部特征提取，对时间序列中局部错位和形态变化的显式建模能力相对有限。PDAN 通过样本与类别原型之间的动态对齐计算，使模型能够更好地适应不同样本之间的局部形态变化，因此在多数数据集上仍然取得了较好的分类结果。

与集成方法相比，PDAN 同样表现出较强鲁棒性。相较于 EE，PDAN 在 40 个距离型数据集中取得 31 胜、4 平、5 负的结果；相较于 BOSS，PDAN 同样取得 31 胜、4 平、5 负的结果。集成方法通常通过组合多个分类器或多个特征空间来提升整体性能，但模型结构相对复杂，计算和存储成本也更高。PDAN 使用统一的原型动态对齐框架，在不依赖复杂集成结构的情况下，仍能够取得与集成方法相近甚至更优的分类表现。

综合上述结果可以看出，PDAN 在距离型时间序列数据集上表现突出。其相较于 NN-DTW、LSTM、InceptionTime、ROCKET、EE 和 BOSS 均取得了更多胜场，说明所提出的动态原型对齐机制能够有效处理时间序列中的局部错位和形态变化。同时，PDAN 在与深度学习方法和集成方法的比较中也保持了较好的竞争力，表明该方法在分类准确率、模型复杂度和低资源适应能力之间取得了较好的平衡。

3.3. 低资源与完整训练结果

为了进一步分析 PDAN 在不同训练样本规模下的性能表现，本文在 6 个 UCR 数据集上设置了低资源和完整训练实验。实验分别使用 1%、10%和 100%的训练样本进行训练，并与 NN-DTW、InceptionTime、ROCKET 和 LSTM 进行比较。实验结果如表 1 所示。

Table 1. Accuracy on six UCR datasets under different training ratios
表 1. 六个 UCR 数据集在不同训练比例下的准确率

数据集	训练比例	NN-DTW	Inception Time	ROCKET	LSTM	PDAN
ECG5000	1%	0.843	0.729	0.745	0.704	0.839
	10%	0.911	0.853	0.829	0.836	0.918
	100%	0.926	0.944	0.947	0.935	0.958
HandOutlines	1%	0.826	0.720	0.752	0.703	0.834
	10%	0.840	0.857	0.878	0.823	0.865
	100%	0.879	0.956	0.945	0.884	0.894
MidPhalanxOutlineCorrect	1%	0.625	0.696	0.706	0.689	0.665
	10%	0.690	0.758	0.741	0.765	0.754
	100%	0.714	0.820	0.844	0.806	0.823
StarLightCurves	1%	0.839	0.832	0.824	0.820	0.846
	10%	0.885	0.883	0.848	0.853	0.892
	100%	0.910	0.978	0.983	0.973	0.970
Strawberry	1%	0.829	0.733	0.771	0.752	0.814
	10%	0.917	0.830	0.852	0.812	0.932
	100%	0.943	0.982	0.979	0.960	0.988

续表

Wafer	1%	0.885	0.760	0.721	0.709	0.897
	10%	0.924	0.801	0.813	0.807	0.964
	100%	0.982	0.998	0.996	0.994	0.992

从表 1 可以看出, 在 1%训练样本条件下, PDAN 在 HandOutlines、StarLightCurves 和 Wafer 数据集上取得最高准确率, 在 ECG5000 数据集上与 NN-DTW 的结果非常接近。这说明在极低资源条件下, PDAN 能够利用类别原型和动态对齐机制保持较好的分类稳定性。相比之下, InceptionTime、ROCKET 和 LSTM 等深度学习模型在部分数据集上性能下降较明显, 说明这些方法在极少标注样本条件下更容易受到训练数据不足的影响。

在 10%训练样本条件下, PDAN 的优势更加明显。PDAN 在 ECG5000、StarLightCurves、Strawberry 和 Wafer 数据集上取得最高准确率, 并在 HandOutlines 和 MidPhalanxOutlineCorrect 数据集上保持较强竞争力。与 1%训练条件相比, 当可用训练样本数量增加时, PDAN 能够进一步优化特征编码器和类别原型, 使模型性能提升。这表明 PDAN 不仅适用于冷启动场景, 也能够随着标注样本增加而持续学习。

在 100%完整训练条件下, PDAN 仍保持较好的分类性能。在 ECG5000 和 Strawberry 数据集上, PDAN 分别取得 0.956 和 0.986 的准确率, 超过其他对比方法。在 Wafer 和 StarLightCurves 数据集上, 虽然 InceptionTime 或 ROCKET 略优于 PDAN, 但 PDAN 的准确率仍然保持在较高水平。这说明 PDAN 在数据较充分的情况下仍具有竞争力, 并未因采用原型对齐结构而限制模型的学习能力。

综合低资源和完整训练实验结果可以发现, PDAN 在不同训练数据规模下均具有较稳定的表现。在样本极少时, PDAN 受益于原型学习和动态时间对齐机制, 能够获得接近或优于 NN-DTW 的分类效果; 当训练样本增加时, PDAN 又能够通过端到端训练持续优化模型参数, 进一步提升分类性能。因此, PDAN 在一定程度上兼具传统动态时间规整方法的小样本适应能力和神经网络模型的可训练性, 适合用于低资源条件下的时间序列分类任务。

4. 讨论

本文实验结果表明, PDAN 在低资源时间序列分类任务中具有较好的适应能力。与传统 NN-DTW 方法相比, PDAN 并非直接将训练样本作为匹配模板, 而是通过可学习类别原型提取类别代表模式, 从而降低噪声样本和个别异常序列对分类结果的影响。与普通深度学习模型相比, PDAN 显式引入动态时间对齐机制, 能够更好地处理序列中的局部时间偏移和形态变化, 因此在多个距离型时间序列数据集上取得了较好的分类结果。

同时, 本文方法仍存在一定局限。首先, Soft-DTW 对齐计算需要构建样本序列与类别原型之间的局部距离矩阵, 并通过动态规划方式计算累积匹配代价, 因此其计算复杂度通常与序列长度和原型长度相关。当序列长度较长、类别数量较多或每类设置多个原型时, 模型训练和推理时间可能增加。与 NN-DTW 相比, PDAN 需要额外训练特征编码器和类别原型, 但在推理阶段只需将测试样本与类别原型进行匹配, 而不必与全部训练样本逐一比较, 因此在训练样本数量较多时具有一定潜在优势。与 InceptionTime、ROCKET 等深度模型相比, PDAN 的动态对齐模块会带来额外计算开销, 但也增强了模型对局部时间错位的适应能力。本文主要关注分类性能验证, 尚未对不同方法的实际运行时间进行系统比较, 后续研究将进一步补充训练时间、推理时间和模型参数量等指标, 以更全面评估模型的计算效率。

此外, 当前模型主要采用类别原型描述每一类时间序列的典型模式。然而, 在实际任务中, 同一类别内部可能存在多种不同的子模式。例如, 同一类生理信号可能因个体差异、采集环境或状态变化而呈现不

同形态, 仅使用单一类别原型可能难以充分刻画类内多模态分布。针对这一问题, 后续研究可以将 PDAN 扩展为多原型结构, 即为每个类别设置多个可学习原型, 并在分类时选择与输入样本最匹配的原型进行动态对齐。该扩展有助于增强模型对类内差异的表达能力, 提高复杂时间序列分类任务中的适应性。

未来研究可以从三个方面展开: 一是进一步优化动态对齐计算过程, 提高模型训练和推理效率; 二是结合图神经网络或注意力机制建模多变量时间序列中的通道关系; 三是在脑电信号、医学诊断和工业监测等实际任务中开展更多验证, 以进一步评估方法的应用价值。

5. 总结

在本文中, 我们提出了一种基于原型动态对齐网络的时间序列分类方法(PDAN), 并将其应用于低资源条件下的时间序列分类任务。该方法针对传统神经网络模型依赖大量标注样本、在小样本场景下容易过拟合, 以及传统动态时间规整方法难以端到端训练的问题, 将神经网络表示学习、类别原型构建和软动态时间对齐机制进行结合。具体而言, 本文首先利用特征编码模块对原始时间序列进行隐空间表示, 以提取更加稳定的时序特征; 然后构建可学习类别原型, 用于描述不同类别的典型结构模式; 最后通过软动态时间对齐机制计算输入样本与类别原型之间的匹配距离, 并根据对齐结果完成分类任务。

实验结果表明, PDAN 方法在 UCR 公开时间序列数据集上具有较好的分类性能。与 NN-DTW、LSTM、InceptionTime、ROCKET、BOSS 和 Elastic Ensemble 等基线方法相比, PDAN 在多个距离型时间序列数据集上取得了更优或具有竞争力的分类结果。特别是在低资源训练场景下, PDAN 能够有效利用类别原型和动态对齐机制, 在训练样本较少时保持较稳定的分类性能; 在训练样本较充分的情况下, PDAN 仍能够通过端到端训练进一步优化特征表示和类别原型, 保持较好的竞争力。

综上所述, PDAN 方法在一定程度上兼顾了传统动态时间规整方法的小样本适应能力和神经网络模型的可训练性。该方法能够有效处理时间序列中的局部时间错位问题, 并通过可学习原型提升类别表示能力, 为低资源时间序列分类任务提供了一种可行思路。后续研究将从三个方面进一步改进: 一是补充训练时间、推理时间和参数量等指标, 以系统分析模型计算效率; 二是探索每类多个原型的扩展形式, 以增强模型对类内多模态结构的表达能力; 三是在更多实际应用场景中验证模型泛化性能, 进一步提升方法的实用价值。

参考文献

- [1] Bagnall, A., Lines, J., Bostrom, A., Large, J. and Keogh, E. (2017) The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances. *Data Mining and Knowledge Discovery*, **31**, 606-660. <https://doi.org/10.1007/s10618-016-0483-9>
- [2] Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.M., Zhu, Y., Gharghabi, S., et al. (2019) The UCR Time Series Archive. *IEEE/CAA Journal of Automatica Sinica*, **6**, 1293-1305. <https://doi.org/10.1109/jas.2019.1911747>
- [3] Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M. and Bagnall, A. (2020) The Great Multivariate Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances. *Data Mining and Knowledge Discovery*, **35**, 401-449. <https://doi.org/10.1007/s10618-020-00727-3>
- [4] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L. and Muller, P. (2019) Deep Learning for Time Series Classification: A Review. *Data Mining and Knowledge Discovery*, **33**, 917-963. <https://doi.org/10.1007/s10618-019-00619-1>
- [5] Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [6] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., et al. (2020) Inceptiontime: Finding Alexnet for Time Series Classification. *Data Mining and Knowledge Discovery*, **34**, 1936-1962. <https://doi.org/10.1007/s10618-020-00710-y>
- [7] Dempster, A., Petitjean, F. and Webb, G.I. (2020) ROCKET: Exceptionally Fast and Accurate Time Series Classification Using Random Convolutional Kernels. *Data Mining and Knowledge Discovery*, **34**, 1454-1495. <https://doi.org/10.1007/s10618-020-00701-z>

-
- [8] Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A. and Bagnall, A. (2021) HIVE-COTE 2.0: A New Meta Ensemble for Time Series Classification. *Machine Learning*, **110**, 3211-3243. <https://doi.org/10.1007/s10994-021-06057-9>
 - [9] Wang, Z., Yan, W. and Oates, T. (2017) Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. 2017 *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, 14-19 May 2017, 1578-1585. <https://doi.org/10.1109/ijcnn.2017.7966039>
 - [10] Craik, A., He, Y. and Contreras-Vidal, J.L. (2019) Deep Learning for Electroencephalogram (EEG) Classification Tasks: A Review. *Journal of Neural Engineering*, **16**, Article ID: 031001. <https://doi.org/10.1088/1741-2552/ab0ab5>
 - [11] Faust, O., Hagiwara, Y., Hong, T.J., Lih, O.S. and Acharya, U.R. (2018) Deep Learning for Healthcare Applications Based on Physiological Signals: A Review. *Computer Methods and Programs in Biomedicine*, **161**, 1-13. <https://doi.org/10.1016/j.cmpb.2018.04.005>
 - [12] Berndt, D.J. and Clifford, J. (1994) Using Dynamic Time Warping to Find Patterns in Time Series. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Seattle, 31 July-1 August 1994, 359-370.
 - [13] Keogh, E. and Ratanamahatana, C.A. (2005) Exact Indexing of Dynamic Time Warping. *Knowledge and Information Systems*, **7**, 358-386. <https://doi.org/10.1007/s10115-004-0154-9>
 - [14] Jeong, Y.S., Jeong, M.K. and Omitaomu, O.A. (2011) Weighted Dynamic Time Warping for Time Series Classification. *Pattern Recognition*, **44**, 2231-2240. <https://doi.org/10.1016/j.patcog.2010.09.022>
 - [15] Lines, J. and Bagnall, A. (2015) Time Series Classification with Ensembles of Elastic Distance Measures. *Data Mining and Knowledge Discovery*, **29**, 565-592. <https://doi.org/10.1007/s10618-014-0361-2>
 - [16] Abanda, A., Mori, U. and Lozano, J.A. (2019) A Review on Distance Based Time Series Classification. *Data Mining and Knowledge Discovery*, **33**, 378-412. <https://doi.org/10.1007/s10618-018-0596-4>
 - [17] Snell, J., Swersky, K. and Zemel, R. (2017) Prototypical Networks for Few-Shot Learning. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 4080-4090.
 - [18] Cuturi, M. and Blondel, M. (2017) Soft-DTW: A Differentiable Loss Function for Time-Series. *Proceedings of the 34th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 894-903.
 - [19] Schäfer, P. (2015) The BOSS Is Concerned with Time Series Classification in the Presence of Noise. *Data Mining and Knowledge Discovery*, **29**, 1505-1530. <https://doi.org/10.1007/s10618-014-0377-7>