This paper has been accepted for publication and will appear in the Machine Learning Journal.

Time Series Representations Classroom (TSRC): A Teacher-Student-based Framework for Interpretability-enhanced Unsupervised Time Series Representation Learning

Wadie Skaf [1*], Mitra Baratchi [2] and Holger Hoos [1,2]

[1*]Chair for Artificial Intelligence Methodology (AIM), RWTH Aachen University, Theaterstraße 35-39, Aachen, 52062, Nordrhein-Westfalen, Germany.

[2]Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Niels Bohrweg 1, Leiden, 2333 CA, South Holland, The Netherlands.

*Corresponding author(s). E-mail(s): skaf@aim.rwth-aachen.de;
Contributing authors: m.baratchi@liacs.leidenuniv.nl;
hh@aim.rwth-aachen.de;

Abstract

Time series representation learning is the process of extracting condensed and meaningful representations from raw sequential data, with unsupervised representation learning offering methods to do so without the need for labelled data. Reconstruction-based deep-learning methods are capable of deriving representations from sequential data in an unsupervised setting and offer enhanced interpretability due to their capability of decoding extracted representations; however, these methods often fall short of contrastive-based methods regarding the quality of representations, as the latter utilise contrastive learning to produce representations that are as close as possible in the embedding space for similar samples and far apart for dissimilar ones. We propose Time Series Representations Classroom (TSRC), a framework that leverages knowledge distillation and curriculum learning to combine the interpretability of reconstruction-based methods with the capabilities of contrastive-based methods. This framework consists of a hybrid loss function that combines reconstruction and contrastive losses and a curriculum that guides the learning process. We compare the performance of methods trained within the TSRC framework using the downstream task of time

series clustering on 112 datasets from the UCR Archive against the same methods trained without the TSRC framework and 4 baselines from the literature. Our empirical results demonstrate that methods trained within the TSRC framework deliver better results compared to the same methods trained without it, achieving higher average rankings between 6.88% and 17.47% in external cluster evaluation and between 62.15% and 75.07% in internal cluster evaluation. Furthermore, the results demonstrate that models trained using the TSRC framework produce representations that are more transferable, achieving, without additional tuning, on average 14.02% higher average rankings in time series classification compared to the same models trained without the TSRC framework.

Keywords: Representation Learning, Time Series, Contrastive Learning, Knowledge Distillation, Curriculum Learning, Explainable Artificial Intelligence (XAI)

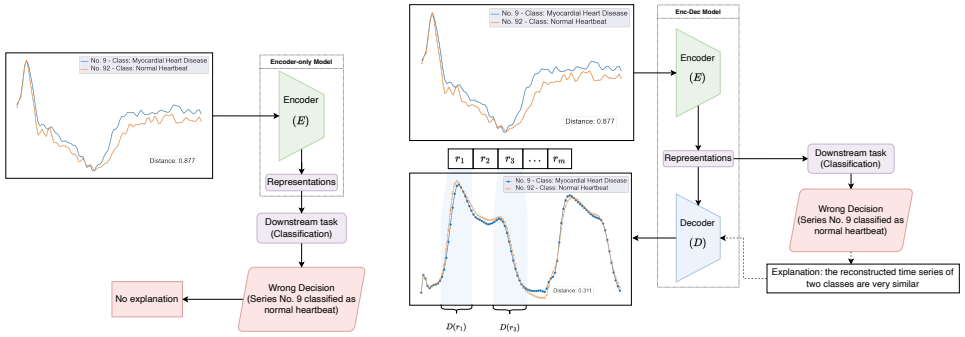
1 Introduction

Time series analysis forms a prominent research area, with applications in various downstream tasks, such as forecasting (De Gooijer and Hyndman, 2006), clustering (Ma et al, 2019, 2021), classification (Ismail Fawaz et al, 2019), and anomaly detection (Skaf and Horváth, 2022), as well as various domains, including finance (Sezer et al, 2020), environmental science (Chen et al, 2018), and healthcare (Skaf et al, 2023).

Unsupervised time series representation learning has proven to be an efficient approach for distilling and extracting useful features from complex raw time series data, enabling transfer learning to many downstream tasks given that the learned representations can capture potentially valuable information within a time series (Meng et al, 2023; Zhang et al, 2024a). To be effective in a diverse range of tasks and applications, representations should have the following two important characteristics: (1) They should effectively preserve the rank order of similarities between instances in the embedding space. In other words, representations of similar time series samples should be as close as possible in the embedding space, while representations of samples that are not similar to each other should be as far as possible in the embedding space. (2) Representations should be as interpretable as possible. In particular, it should be possible to determine which element of the representations corresponds to which section of the time series data.

Two commonly used strategies for representation learning are reconstruction- and contrastive-based methods (Meng et al, 2023; Zhang et al, 2024a). Reconstruction-based methods, also referred to as encoder-decoder (Enc-Dec) methods, operate by jointly training an encoder that extracts representation from an input time series signal and a decoder that works on reconstructing the signal from the extracted representation with the objective of minimising the error between the original and reconstructed signals; the trained encoder can then be used to extract representations from time series samples. Reconstruction-based methods commonly employ Recurrent Neural Networks (RNNs) encoder-decoder architectures, which are natural choices for modelling sequential data. However, RNNs (including GRU and LSTM variants) often require significant amounts of training data to capture the underlying patterns in time series (Goodfellow et al, 2016), while missing the ability to correctly capture the similarity between instances and their orderings in the embedding space. These methods can offer greater interpretability by having decodable representations. -based methods are inherently interpretable; this interpretability comes from the possibility of visualising and studying which segment of the time series is related to a specific element of the representation and to observe what the model has learned during the training process in general (Le Naour et al, 2023)—this is generally done by comparing the original and the reconstructed signals (see Figure 1b).

Contrastive-based methods, typically implemented using an encoder-only architecture (Zhang et al, 2024a), utilise the concept of contrastive learning. The model learns by having a positive example and a negative example to generate representations that are as close as possible in the embedding space for positive examples and as far as possible for negative examples using a triplet loss function (see Section 3.1). While these methods stand out for their ability to preserve the similarity between instances in the



(a) Encoder-only Methods (b) Encoder-decoder Methods

Fig. 1: Comparing encoder-only and encoder-decoder methods regarding interpretability for two time series with similar magnitude of different classes (normal heartbeat versus myocardial infraction), taken from the ECG200 dataset that is part of the UCR Archive. The representations of these two samples were extracted using the two types of models, and a downstream task (classification) was performed using a dedicated algorithm. (a) An incorrect decision made in the downstream task (classifying both samples to be of the same class of normal heartbeat) that uses representations extracted from an encoder-only method cannot be explained. (b) Encoder-decoder methods can decode the representations and explain the decision made using its extracted repersentation; in this example, reconstructing the two time series using the decoder shows inaccuracies resulting in time series that are more similar than the original ones ($d_{\hat{s}_9,\hat{s}_{92}} = 0.311 < d_{s_9,s_{92}} = 0.877$)—clearly, the model did not learn to accurately reconstruct the time series nor to distinguish between them properly. The same process of checking the reconstructed samples against the original samples can be used to explain decisions in case of other downstream tasks (such as clustering).

embedding space, they lack interpretability. This is a natural limitation of encoder-only architectures, where there is no reverse method to decode the representation from the embedding space into the original sample space. Figure 1 compares encoder-only and encoder-decoder (Enc-Dec) methods and illustrates how Enc-Dec methods can be beneficial in explaining decisions made based on the extracted representations as well as in assessing what the model has learned.

To achieve representations that satisfy both of the aforementioned characteristics, it would be advantageous to combine the strengths of both contrastive- and reconstruction-based methods while mitigating their limitations, potentially by learning representations through contrastive learning in combination with a decoder that can be used to assess the learned model and interpret the decisions made based on the extracted representations. To accomplish this goal, we propose the Time Series Representations Classroom (TSRC) framework that utilises the principles of knowledge distillation (Hinton et al, 2015) – a technique for transferring knowledge from a large, complex teacher model to a smaller, less-complex student model – and curriculum learning (Bengio et al, 2009) – an approach that first learns simpler tasks

and then gradually introduces more complex tasks. This dual approach leverages the robust, detailed knowledge from a complex teacher model and the structured, progressive training of curriculum learning, potentially improving the ability of the student model to learn better representations.

Our proposed TSRC framework consists of two models: a teacher model and a student model. The teacher model is first trained individually and then used to assist in training the student model by providing hints during training in the form of a hint loss, which quantifies how far the representations generated by the student model are from the representations generated by the teacher model. The main goal of the TSRC framework is to combine the power of a teacher model with a reconstruction-based student model, with the aim of enabling the latter to achieve better results in downstream tasks by improving on its weaknesses. The main contributions of this study are as follows:

1. Developing a novel framework for unsupervised time series representation learning that combines the benefits of contrastive- and reconstruction-based time series representation learning methods.
2. Forming a hybrid loss function that combines both contrastive and reconstruction losses to learn time series representations.
3. Constructing a curriculum tailored for time series representation learning that utilises the designed loss function to train a student model using hints from a teacher model.
4. Evaluating the performance of the framework using combinations of 2 contrastive-based and 2 reconstruction-based methods on 112 widely-used datasets.

The remainder of this article is structured as follows: First, we discuss related work in Section 2; next, we describe the proposed framework in Section 3, loss function in Section 3.1 and curriculum in Section 3.2, followed by the experimental setup in Section 4; after that, we report and summarise our major empirical results in Section 5, and finally discuss limitations and potential future work in Section 7.

2 Related Work

In this section, we review the existing literature on knowledge distillation and curriculum learning and discuss their application in various domains. We then review the existing literature on unsupervised time series representation learning methods and discuss their different types and categories.

2.1 Knowledge Distillation and Curriculum Learning

Knowledge distillation, introduced by Hinton et al (2015), is a technique that involves transferring knowledge from a large, complex teacher model to a smaller, less-complex student model, with the goal of mimicking the behaviour of the teacher model; it mainly works by transferring knowledge through soft labels or intermediate representations. Curriculum learning, introduced by Bengio et al (2009), is a technique inspired by the human learning process, where simpler tasks are learned first, then more complex tasks are gradually introduced; it mainly works by organising training data and tasks in a meaningful order to achieve this gradual increase in task complexity during

training. Both techniques have been predominantly used in domains such as computer vision, reinforcement learning, and natural language processing (Habib et al, 2023; Liu et al, 2023).

In time series analysis, these techniques have been used to a lesser extent. Qiao et al (2023) introduced a distillation-based method that uses Soft-Dynamic Time Warping (Soft-DTW) for class-incremental learning on multivariate time series, achieving improvements in stability and performance. Ay et al (2022) investigated knowledge distillation for fully convolutional networks in time series classification, obtaining results similar to bigger models at reduced computational costs. Zhang et al (2024b) proposed a Cross Reconstruction Transformer (CRT) for self-supervised time series representation learning that utilises curriculum learning to gradually increase the complexity of the training process by increasing the sample dropping ratio, resulting in improved representation quality and robustness. The combined use of knowledge distillation and curriculum learning has not been well explored, with only a few studies reported in the literature. One such example in the vision domain is by Zhao et al (2021), who proposed a curriculum learning knowledge distillation framework for instance-level sequence learning. This framework utilises a teacher-student architecture, where the student model learns from easier picture instances at the beginning, then progresses to more complex examples as its performance increases, all under the supervision of a pre-trained teacher model.

Integrating knowledge distillation and curriculum learning into time series representation learning involves several difficulties, such as handling various neural network architectures and balancing loss functions specific to time series downstream tasks. To the best of our knowledge, the combination of ideas from knowledge distillation and curriculum in time series representation learning has not been previously explored, which makes our TSRC framework the first significant effort in this direction.

2.2 Unsupervised Time Series Representation Learning

Time series representation learning is conceptualised as a transformative function f that maps a specific segment A^j of a time series $X = (x_1, x_2, \ldots, x_T)$ onto a vector space $\mathbb{R}^m$ as formalised in Equation 1. The segment A^j is defined as a contiguous subsequence of X of size n_j, such that $A^j = (x_j, x_{j+1}, \ldots, x_{j+n_j-1}) \in \mathcal{T}^{n_j}$, where $1 \leq j < j + n_j - 1 \leq T$ and $\mathcal{T}^{n_j} \coloneqq \left\{(x_1, x_2, \ldots, x_{n_j}) \in \mathbb{R}^{n_j} \;\middle|\; x_1 \prec x_2 \prec \cdots \prec x_{n_j}\right\}$ is the space of time series segments of length n_j (where $\prec$ denotes that the elements are strictly ordered in time, i.e., each element occurs chronologically after the previous one).

$$f : \mathcal{T}^{n_j} \to \mathbb{R}^m, \quad f(A^j) = \mathbf{v} = (v_1, v_2, \ldots, v_m) \in \mathbb{R}^m \tag{1}$$

According to the taxonomy provided by Meng et al (2023), unsupervised representation learning for time series can be split into multiple categories, namely: (i) deep clustering, (ii) reconstruction-based, and (iii) self-supervised learning methods (Längkvist et al, 2014; Eldele et al, 2021; Yue et al, 2022).

Deep Clustering Methods. These methods aim to combine clustering and representation learning by jointly learning representations and clustering assignments, enabling iterative optimisation of a clustering-oriented objective to learn mappings

from an input space to a new latent space. The main benefit of this category of methods is that the clustering results can be used as pseudo-labels to guide the learning process towards more meaningful representation. Examples of these methods are Deep Temporal Clustering Representation (DTCR) (Ma et al, 2019) and Clustering Representation Learning on Incomplete Time Series Data (CRLI) (Ma et al, 2021). As these methods incorporate a clustering algorithm during training, it makes them reliant on its performance and potentially biased towards the cluster shapes it favours. The aim of our research is to improve the rank order of similarities between instances in the embedding space while maintaining interpretability by having a decoder, rather than just improving clustering performance by making the representations form specific shapes in the embedding space; therefore, this category of methods is not considered.

Reconstruction-based Methods. These methods aim to minimise the difference between the reconstructed output and the raw input, to increase the ability of the model to concentrate on key elements while filtering out irrelevant or noisy information. They generally employ an Enc-Dec architecture (Meng et al, 2023). A prominent example of such a method is TimeNet (Malhotra et al, 2017), which uses a sequence autoencoder (SAE) network based on the Sequence-to-Sequence (Seq2Seq) model (Sutskever et al, 2014) to transform variable-length time series into fixed-dimensional representations. Deconv (Song et al, 2020) is another example; it employs deconvolutional networks to reconstruct time series data by performing inverse convolution and pooling operations, enabling the reconstruction of hidden representations within the network. In our research, we selected student models from this category because they offer interpretability through the possibility of decoding representations.

Self-Supervised Learning Methods. These approaches differ from fully unsupervised methods described earlier by designing diverse pretext tasks that automatically generate useful pseudo-labelled data from the original data. They utilise the inherent structure of data to produce labels, which enables a model to learn meaningful representations without manual annotation. This category of methods is further divided into adversarial, predictive, and contrastive methods, each with advantages and drawbacks (Meng et al, 2023).

Adversarial methods use the problem of distinguishing between real and fake data as a pretext task to learn robust representations for time series. These methods typically establish a two-player mini-max game, in which the generator works to improve its ability to confuse the discriminator and the discriminator aims to become better at identifying real data from fake data. By iteratively training the generator and discriminator in an adversarial manner, these methods allow the generator to learn representations that capture the key characteristics of raw data. Examples of these methods are TimeGAN (Yoon et al, 2019) and MAD-GAN (Li et al, 2019). Adversarial methods were not considered in this research, as training adversarial methods can be unstable and complex, and they require significant computational resources and data (Goodfellow et al, 2016), which makes them unsuitable for our framework—the teacher model in the framework must be stable and reliable to effectively guide the student model.

Predictive methods use tasks such as predicting future, missing, or contextual information as pretext tasks to predict partial data based on limited views; they learn

representations by predicting future or mixed values of partial time series, predicting whether time windows are sampled from the same temporal context, or predicting cross-view representations of original samples. Examples of these methods include TST (Zerveas et al, 2021) and EEG-SSL (Banville et al, 2019). These approaches were not considered in this research, because predictive methods often involve designing specific pretext tasks that may not generalise well across different applications and downstream tasks—tasks such as future value or temporal context prediction are highly dependent on the nature of the time series data and specific downstream tasks. Moreover, the necessity of having these auxiliary tasks as an integral part of the training process—which is usually achieved using a decoder—makes it unfeasible to integrate an additional decoder for interpretability, which is a key goal of this research.

Contrastive methods utilise pretext tasks specificallyy designed to learn useful representations by solving problems that mimic the structure or context of real tasks without requiring labels (Zhang et al, 2024a). This is achieved by generating augmented views of raw data through various transformations and then formulating self-discrimination tasks through contrasting positive and negative instances, where positive and negative instances refer to similar and dissimilar samples, respectively. This approach embraces the underlying similarity across samples instead of directly modelling the complex raw data, eliminating the need for reconstructing the entire output and allowing for the detection of contextualised underlying causes of variation (Mohamed et al, 2022). Contrastive methods balance data augmentation techniques and architectural innovations to improve time series representations (Meng et al, 2023; Zhang et al, 2024a). One example of such methods is Mixing Up Contrastive Learning (MCL) (Wickstrøm et al, 2022), which employs a data augmentation approach that generates new samples by combining two data samples with a mixing component; the label smoothing-motivated pretext task is to predict the intensity of the mixing component based on the two data samples and the augmented sample. A second example is TS2Vec (Yue et al, 2022) which uses multiscale contextual information with granularities to differentiate between samples and generate different views (positive and negative) through masking operations that randomly mask out some time steps; this allows the learning of timestamp-level representations while simultaneously supporting instance-level representations of the full time series via a max pooling technique across timestamps. Time-Contrastive Learning (TCL) (Hyvarinen and Morioka, 2016) is another example, which derives representations using a multilayer perceptron (MLP) in a nonlinear independent component analysis (ICA) model from uniform segments of time series by distinguishing between these segments based on their representations. The main advantage of contrastive methods is that they preserve the order of similarities between instances in the embedding space. These methods typically employ either an encoder-only architecture or incorporate an additional decoder to perform auxiliary tasks, such as forecasting (Zhang et al, 2024a). Neither of these strategies allows mapping from the embedding space back to the original space, which makes these approaches inherently uninterpretable.

Other methods adopted different pretext tasks for self-supervised learning (Foumani et al, 2024a) as opposed to using adversarial, predictive or contrastive tasks. One such method is Series2Vec (Foumani et al, 2024b), which learns representations

by predicting pairwise similarities between series in both the temporal and spectral domains instead of using direct similarity measurements between positive and negative pairs (which is used in contrastive tasks). Another example is Pre-trained HInception-Time (PHIT) (Ismail-Fawaz et al, 2024), which generates flexible convolution filters that can be applied across diverse datasets by identifying the originating dataset of each time series sample. The primary goal of our work presented here is different from that of these approaches; we aim to improve the rank ordering of representations that is achievable using contrastive approaches. This rank ordering is a fundamental quality that is achievable when the similarity between pairs of time series is measured (typically using contrastive learning). While the previously mentioned pretext tasks are novel, the ability of methods such as Series2Vec and HInceptionTime to preserve rank ordering remains unproven; therefore, these methods were not considered in this study.

The goal of this study is to combine the interpretability of reconstruction-based methods with the order-preservation quality of contrastive-based methods. Rather than proposing a new representation learning algorithm, we propose TSRC, a framework for combining existing algorithms from these two categories. Our framework allows the use of a contrastive-based model as a teacher and a reconstruction-based architecture as a student model.

3 Proposed Framework

Our proposed TSRC framework is designed with the aim of combining the power of two classes of models: a contrastive- and a reconstruction-based model, by offering a loss function that combines the abilities of contrastive and reconstruction losses so that contrastive loss can accommodate having a decoder. The TSRC framework follows the principles of knowledge distillation (Hinton et al, 2015) by utilising the multistage distillation-based training approach and additionally integrating curriculum learning (Bengio et al, 2009).

The TSRC framework consists of two encoders: one acting as a teacher and one as a student, and a separate decoder that reconstructs the output of the student encoder, as illustrated in Figure 2. The teacher encoder can be from any category of time series unsupervised representation learning methods, encoder-only, or Enc-Dec models, provided that representations can be generated using only the encoder after training. The student model, however, must be a reconstruction-based model, as this framework requires the student to have a decoder beside the encoder.

The hinting process from the teacher model to the student model, which follows a designed curriculum, is at the heart of this framework. This is performed in an iterative manner by comparing the representation of the input time series extracted by the teacher and student models. The difference between these representations serves as a hint provided by the teacher to the student model. In each iteration, this hint is provided in the form of a loss value based on a set of training samples following a defined curriculum that determines the learning goal. This leads to the key components of this framework: (i) a loss function that allows for joint unsupervised training of two

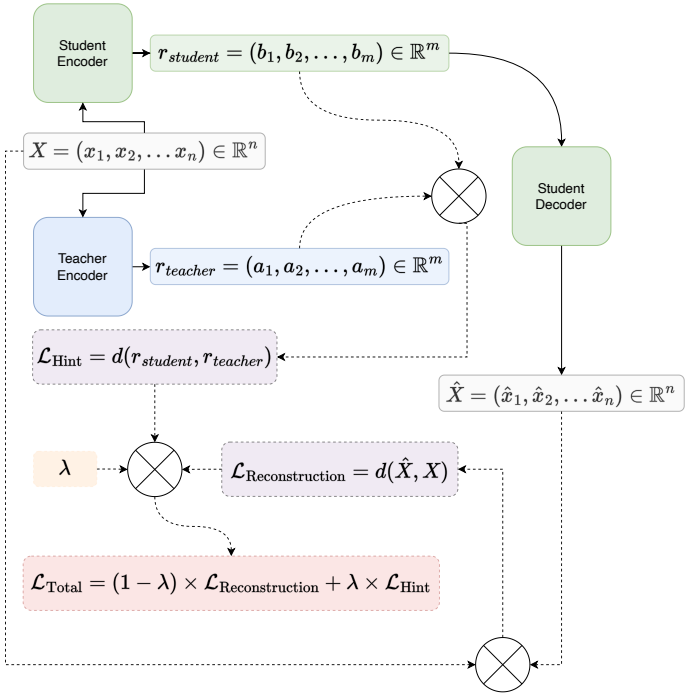


Fig. 2: An illustration of the flow of TSRC framework process. Given a time series sample X, representations $r_{teacher}$ and $r_{student}$ are extracted using a teacher and student encoder, respectively, and a decoder is used to produce a reconstructed sample $\hat{X}$. After that, two losses, a hint loss and a reconstruction loss, are calculated and joined using a coefficient λ. The combination nodes represent the process of adding the components together according to the formula shown in the output component. The "sum nodes", shown as circles with an X inside ($\otimes$), represent an aggregation of input components according to the equation displayed in the output component.

models by integrating reconstruction loss and hint loss to balance representation learning and signal reconstruction (further discussed in Section 3.1) and (ii) a curriculum that controls the process of progressively introducing the hint loss to the total loss along the training epochs; the curriculum is implemented in the form of a number of hyperparameters that are set before the training (further discussed in Section 3.2).

The remainder of this section outlines the loss function and the curriculum. In addition, we explain the process of introducing what we call a hint during training and its effect on the total loss.

3.1 Loss Function

In this section, we provide background information on contrastive loss for time series representation learning. We then discuss two methods of combining contrastive and

reconstruction losses, detailing the rationale behind the method selected for use in the TSRC framework.

Contrastive Loss for Time Series Representation Learning. This loss function is commonly defined based on the idea of a triplet of samples: an anchor sample, a positive sample that is similar to the anchor, and a negative sample that is dissimilar from the anchor; positive and negative samples can be obtained using multiple methods in an unsupervised manner (without using labelled data). The most popular approach for creating positive and negative samples is data augmentation through hand-crafted transformations, such as noise injection, window slicing, and window wrapping rules (Zhang et al, 2024a). The goal of contrastive learning is to ensure that the positive sample is closer to the anchor than the negative sample, by at least a certain margin; this forms a triplet loss. Deep learning-based methods that utilise contrastive learning in time series representation learning commonly adopt an encoder-only architecture (Zhang et al, 2024a), where this encoder is trained by generating embeddings ($z = E(X)$) of the series X. Equation 2 presents the contrastive loss for time series representation learning.

$$\mathcal{L}_{\text{Contrastive}} = \max(d(z_a, z_p) - d(z_a, z_n) + \delta, 0), \tag{2}$$

where z_a, z_p, z_n are the embeddings of the anchor, positive and negative time series samples, respectively, $d(\cdot)$ is the distance metric, such as Euclidean distance, and δ is a hyperparameter that defines the minimum desired gap between the distance from the anchor to the positive sample and the distance from the anchor to the negative sample. When $d(z_a, z_p) - d(z_a, z_n) + \delta \leq 0$, the triplet is considered "correctly ordered" with respect to the margin, leading to zero, which indicates that the model does not require adjustments for this particular triplet; otherwise, the model adjusts its weights to produce embeddings that reduce this difference in future iterations.

In order to accommodate a decoder in combination with contrastive loss, in the following sections, we propose to reconstruct the resulting representations based on a reconstruction loss while training within the TSRC framework.

3.1.1 Naïve Hybrid Loss Training

One possible way to expand the contrastive loss function $\mathcal{L}_{\text{Contrastive}}$ is by directly adding a reconstruction loss part, as shown in Equation 3.

$$\begin{aligned}\mathcal{L}_{\text{Total}} &= (1-\lambda) \cdot \left(\frac{1}{\alpha}\right) \cdot \mathcal{L}_{\text{Reconstruction}} + \lambda \cdot \left(\frac{1}{\beta}\right) \cdot \mathcal{L}_{\text{Contrastive}} \\ &= \lambda_1 \cdot \mathcal{L}_{\text{Reconstruction}} + \lambda_2 \cdot \mathcal{L}_{\text{Contrastive}},\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{Reconstruction}} = d(\hat{X}, X)$ is the reconstruction loss according to a distance metric $d(\cdot)$, such as Euclidean distance, $\lambda_1 = (1-\lambda) \cdot \frac{1}{\alpha}$ and $\lambda_2 = \lambda \cdot \frac{1}{\beta}$ are the effective weights for the reconstruction and contrastive loss functions, respectively, and α and β are normalisation factors that aim to balance the contributions of each loss component that has a different value range.

However, combining the two losses (contrastive and reconstruction losses) as introduced in Equation 3 is not meaningful. The two components of the loss function are

nonhomogeneous, with each having its own range of values. The contrastive loss is calculated based on the values of the embeddings using a triplet loss, as shown in Equation 2, which measures the difference between two distances, while the reconstruction loss is calculated based on the values of the original and reconstructed sample, X and $\hat{X}$, respectively, using a distance metric $d(\hat{X}, X)$. To ensure that the loss function combining reconstruction and contrastive loss is meaningful, the following needs to be considered:

1. Determining α and β: These hyperparameters are critical, because they reduce the range of each loss term to a uniform scale in an attempt to make them homogenous (see Equation 3). One way to compute α and β is by using the statistical characteristics of the losses over a validation dataset; for example, α might be determined as the average or maximum value of the contrastive loss measured during the preliminary phase, and the same for β: $\alpha = \mathbb{E}[\mathcal{L}_{\text{Reconstruction}}]$, $\beta = \mathbb{E}[\mathcal{L}_{\text{Contrastive}}]$. However, this approach relies on the assumption that these statistics are stable and consistent, which might not hold, as the value of distance metrics can span a very wide range, rendering the choice of suitable values for these parameters difficult.
2. Dynamic normalisation of α and β: The contrast in loss values (i.e., the relative difference or ratio between the reconstruction loss and the hint loss) can change during training. During training, the contrast might fluctuate as the values of the losses change in relation to changes in the weights of the model, leading to the additional requirement of dynamically modifying the normalisation parameters α and β; this in turn leads to additional computing cost.
3. Potential conflicts between loss functions: Even with successful determination of initial values of α and β and dynamically modifying them during training, optimising these two loss functions at the same time can lead to them contradicting each other (unbalanced training). For example, given an encoder-decoder method, this contradiction could occur because the contrastive loss pushes the embeddings to be properly distant in the representation space based on their similarities, while the reconstruction loss pushes for minimal distortion in the reconstructed output; this joint objective can cause oscillations during training, preventing convergence to an optimal solution for both the encoder and the decoder. Moreover, while there are models in the literature (Meng et al, 2023; Zhang et al, 2024a) in which contrastive and reconstruction losses are utilised individually (i.e., contrastive- and reconstruction-based methods), there is no guarantee or evidence that using them together simultaneously will train encoder-decoder methods successfully.

3.1.2 Multi-Stage Distillation-based Training

In this section, we address the three aforementioned issues by proposing another approach for combining contrastive and reconstruction losses through multi-staged training, which can be formulated as shown in Equation 4.

$$\begin{aligned} \mathcal{L}_{\text{Total}} &= (1-\lambda)\cdot\mathcal{L}_{\text{Reconstruction}} + \lambda\cdot\mathcal{L}_{\text{Hint}} \\ &= (1-\lambda)\cdot d(\hat{X}, X) + \lambda\cdot d(r_{\text{student}}, r_{\text{teacher}}), \end{aligned} \tag{4}$$

where $r_{teacher}$ and $r_{student}$ denote the representations generated by the teacher and

student models, $\mathcal{L}_{\text{Hint}} = d(r_{student}, r_{teacher})$ is the hint loss according to a distance metric $d(\cdot)$ such as the Euclidean distance, and λ is the weight of the hint loss.

In the first stage of training, a pre-trained encoder is obtained, which is used as a teacher model in the second stage of training. In the second stage, knowledge is distilled (Hinton et al, 2015) from this pre-trained encoder (teacher) to an encoder-decoder model (student), so the latter learns how to extract embedding under the supervision of the teacher model. In this approach, the loss function consists of a hint loss that quantifies the difference between the representations produced by the student encoder and the ones produced by the teacher encoder for the same sample, and a reconstruction loss that quantifies the difference between the reconstructed and original samples. This approach has the following advantages over that previously mentioned in Section 3.1.1:

1. The two components of the loss function are homogeneous (Equation 4), which eliminated the need to estimate additional normalisation factors; in this case, there is neither computing overhead nor any additional difficulties regarding dynamically normalising these factors during training.
2. There is only one hyperparameter to set, λ, which is used to determine the proportion of each loss component in the total loss. A single hyperparameter is sufficient because both sub-losses are homogeneous as a result of being calculated using the same distance metric, $d(\cdot)$. This hyperparameter can either remain constant throughout the training or can be changed following a given schedule, a curriculum inspired by the concept of curriculum learning (Bengio et al, 2009), which offers more flexibility and capacity to handle different cases of teacher and student models with different performance differences. The curriculum is designed by setting multiple hyperparameters (this is discussed in more detail in Sections 3.2 and 5.1).
3. There are no concerns about the two losses, contrastive and reconstruction losses, contradicting each other, as each of them is trained individually. In addition, the methods used to combine these losses (i.e., knowledge distillation (Hinton et al, 2015) and curriculum learning (Bengio et al, 2009)) have been used in the literature for several years and have demonstrated success in multiple domains. This minimises the potential for conflicts or other issues during training.

3.2 Curriculum

The TSRC framework encompasses a curriculum, which is the method based on which the knowledge from the teacher model is introduced to the student model. The curriculum in principle controls how the loss function is dynamically calculated during training, i.e., controlling the λ hyperparameter shown in Equation 4. This coefficient determines the ratio of each of $\mathcal{L}_{\text{Reconstruction}}$ and $\mathcal{L}_{\text{Hint}}$ in calculating the total loss $\mathcal{L}_{\text{Total}}$ during the training process.

To emphasise the difference between parameter λ while training and validation, we use λ_{train} and $\lambda_{validation}$ (see Equation 5). The importance of these components stems from the fact that potential teacher and student models vary in performance, and therefore, a curriculum can be designed to obtain the best possible outcome by controlling the dynamic changes of λ_{train} during the training process based on the

initial performance difference between the teacher and the student model; i.e., when the teacher is significantly better than the student, we aim to start with a large λ_{train} to give a higher weight to the hint loss at the beginning of the training process; conversely, when the performance difference between the student and the teacher is not significant, we aim to start with a low λ_{train} to give a higher weight to the reconstruction loss. While λ_{train} changes during the training process, $\lambda_{validation}$ is set to remain constant to ensure a consistent evaluation. $\lambda_{validation}$ is a design choice with no impact on the training procedure, as its sole purpose is to identify and save the "best" model throughout the training process, where the "best" model refers to the student model that learnt the most from the teacher model during training according to the loss function. $\lambda_{validation}$ can be set to any number in the range $[0, 1]$. When it is set to 0, the validation favours models with the best reconstruction loss (i.e., a model with representations that can be best reconstructed by the student decoder), and when it is set to 1, it favours models with the best contrastive loss (i.e., a model with representations that are as close as possible to the teacher's representation). Setting $\lambda_{validation}$ to values between 0 and 1 leads to a mix of both objectives. $\lambda_{validation}$ becomes more important when the goal is to extract representations using the encoder from the "best" model identified through training and can be ignored when the goal is to extract representations from the final model that is saved at the end of the training process.

Designing the curriculum is done by setting all the hyperparameters shown in Equation 5: a, b, ψ_0, ψ_t, and τ. This equation shows how λ_{train} is calculated based on the current training epoch ψ and the hyperparameters of the curriculum, while $\lambda_{validation}$ is a value that remains constant during training.

$$\lambda_{train}(\psi) = \begin{cases} a + (b-a) \cdot \left(\frac{\psi - \psi_0}{\psi_t - \psi_0}\right)^{\tau} & \text{for} \quad \psi \geq \psi_0 \\ 0 & \text{for} \quad \psi < \psi_0 \end{cases}; \qquad \lambda_{validation} \in [0, 1] \tag{5}$$

- a and b denote the starting and final value of λ_{train}, respectively ($a \leq \lambda_{train} \leq b$). Here, $a, b \in [0, 1]$; $a \leq b$.
- ψ_0 denotes the epoch in which the coefficient λ_{train} begins to have values greater than 0 (see Equation 5), and ψ_t denotes the total number of training epochs. Here, $\psi_0 < \psi_t$.
- τ denotes the increment rate according to which λ_{train} increases during the training process (see Equation 5). The increment rate controls how quickly the hint loss weight increases during training, with three main cases for its value (see Figure 3):
  - ▶ $\tau = 1$: the increment is linear. In this case, λ_{train} increases at a constant rate from a to b over the training epochs; the total loss smoothly transitions from being more influenced by the reconstruction loss to being more influenced by the hint loss in a balanced manner.
  - ▶ $\tau > 1$: the increment follows a convex curve shape, starting slow and accelerating over time. In this case, at the beginning of training, λ_{train} increases very

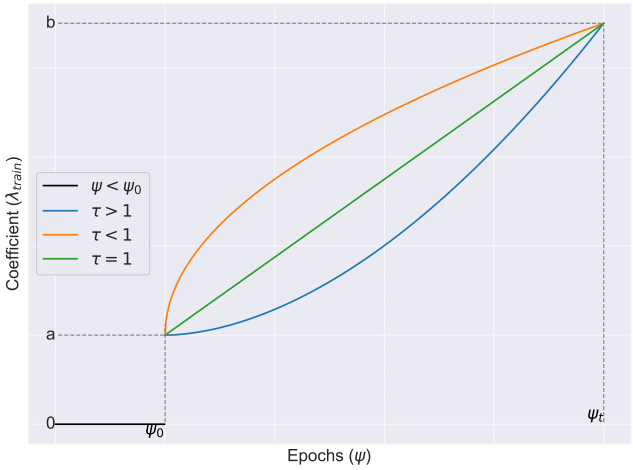


Fig. 3: Changes of the coefficient λ_{train} with respect to the epoch ψ and the hyper-parameters of the designed curriculum (a, b, ψ_0, ψ_t, and τ), considering different scenarios based on the value of the increment rate τ. Here, a and b denote the starting and final values of λ_{train}, respectively; ψ_0 denotes the epoch where the coefficient λ_{train} begins to have values greater than 0; ψ_t denotes the total number of training epoch; and τ denotes the increment rate according to which λ_{train} increases during the training process.

slowly, giving more weight to the reconstruction loss. As training progresses, λ_{train} increases more rapidly, shifting the influence towards the hint loss.

- $\tau < 1$: the increment follows a concave curve shape, starting fast and decelerating over time. In this case, λ_{train} increases rapidly at the beginning, giving substantial weight to the hint loss early in the training process. As training continues, the rate of increase slows, stabilising the influence of the hint loss.

The previously mentioned hyperparameters control the influence of the teacher on the student as well as how the hint loss is introduced during training by dynamically adjusting λ_{train} throughout the training process. These hyperparameters should be set for each experiment based on the goals and the teacher and student models (more details about designing the curriculum for different model combinations are discussed with examples in Section 5.1.) Algorithm 1 illustrates the multi-stage distillation-based training with curriculum and validation used in our experiments. The flow of the curriculum process is illustrated in Figure 2.

Algorithm 1 Multi-Stage Distillation-based Training with Curriculum and Validation

Input: TeacherEncoder model, StudentEncoder model, Decoder model, Curriculum $(a, b, \psi_0, \psi_t, \tau, \lambda_{validation})$, Distance metric $d(\cdot)$, Time series sample X, Validation sample X_{val}

Output: Trained StudentEncoder, Trained Decoder

▷ *Stage 1: Train the teacher encoder using contrastive loss function*

TrainTeacherEncoder()

▷ *Stage 2: train the student encoder using reconstruction loss with hints*

best_loss $\leftarrow \infty$

for ψ from 1 to ψ_t **do**

Training Step:

▷ *Extract representations using the Teacher Encoder*

$r_{\text{teacher}} \leftarrow \text{TeacherEncoder}(X)$

▷ *Extract representations using the Student Encoder*

$r_{\text{student}} \leftarrow \text{StudentEncoder}(X)$

▷ *Reconstruct the signal using the Decoder*

$\hat{X} \leftarrow \text{Decoder}(r_{\text{student}})$

▷ *Compute Hint Loss*

$\mathcal{L}_{\text{Hint}} \leftarrow d(r_{\text{student}}, r_{\text{teacher}})$

▷ *Compute Reconstruction Loss*

$\mathcal{L}_{\text{Reconstruction}} \leftarrow d(\hat{X}, X)$

if $\psi < \psi_0$ **then**

$\lambda_{train} \leftarrow 0$

else

$\lambda_{train} \leftarrow a + (b - a) \cdot \left(\frac{\psi - \psi_0}{\psi_t - \psi_0} \right)^{\tau}$

end if

$\mathcal{L}_{\text{Total}} \leftarrow (1 - \lambda_{train}) \cdot \mathcal{L}_{\text{Reconstruction}} + \lambda_{train} \cdot \mathcal{L}_{\text{Hint}}$

Update(StudentEncoder, Decoder, $\mathcal{L}_{\text{Total}}$)

Validation Step:

$r_{\text{teacher\_val}} \leftarrow \text{TeacherEncoder}(X_{val})$

$r_{\text{student\_val}} \leftarrow \text{StudentEncoder}(X_{val})$

$\hat{X}_{val} \leftarrow \text{Decoder}(r_{\text{student\_val}})$

$\mathcal{L}_{\text{Hint\_val}} \leftarrow d(r_{\text{student\_val}}, r_{\text{teacher\_val}})$

$\mathcal{L}_{\text{Reconstruction\_val}} \leftarrow d(\hat{X}_{val}, X_{val})$

$\mathcal{L}_{\text{Total\_val}} \leftarrow (1 - \lambda_{validation}) \cdot \mathcal{L}_{\text{Reconstruction\_val}} + \lambda_{validation} \cdot \mathcal{L}_{\text{Hint\_val}}$

if $\mathcal{L}_{\text{Total\_val}} <$ best_loss **then**

best_loss $\leftarrow \mathcal{L}_{\text{Total\_val}}$

SaveModel(StudentEncoder, Decoder)

end if

end for

4 Setup of Experiments

This section outlines the datasets, the evaluation procedure, the evaluation metrics, and the baseline methods used in our research study. It provides information on details common to all experiments; individual experiments are motivated and described in Section 5.

4.1 Datasets

In our experiments, we used the UCR Dataset Archive (Dau et al, 2019) with a total number of 112 datasets[1]. For each dataset, 50% of the samples were used for training and 50% for testing; 35% of the training samples were used for validation. The dataset archive is publicly available[2]. A number of datasets (16 datasets) from the archive were excluded, particularly the ones that have varied time series length and the ones that contain missing values (NaN). Datasets with varied lengths were excluded in order to have constant sequence lengths and consistent comparison, while datasets with missing values were excluded because handling missing values adds a robustness requirement for teacher and student models—in this research, we mainly assess the impact of the TSRC framework, leaving the analysis of robustness to future work (see Section 7). Given that the datasets in the UCR Archive are independent of each other, excluding a number of them does not impact the assessment on the remainder. Moreover, the remaining 112 datasets cover a broad range of practical applications, including sensors, health, and other domains, and jointly represent a compelling benchmark.

4.2 Evaluation Procedure

As mentioned in Section 1, representations should preserve the rank order of similarities, i.e., representations of similar time series samples should be as close as possible in the embedding space, while representations of dissimilar samples should be as far as possible in the embedding space. To quantify this, we chose a time series clustering downstream task, which makes it possible to directly assess the ability of a given model to form clusters that mirror similarities and dissimilarities of the samples by using an external clustering evaluation metric.

Following the work of Zhang et al (2019) and Ma et al (2019), we used the Rand Index (RI) parameter for external cluster evaluation; however, instead of RI, we chose to use Adjusted Rand Index (ARI), as the latter is adjusted for chance, providing a more reliable measure of clustering performance in our use case as we are experimenting on more datasets (compared to Zhang et al (2019) and Ma et al (2019)). ARI is an external cluster evaluation metric that compares clustering results to ground truth labels for each sample, assessing how well the clustering obtained from a given model corresponds to those true labels. ARI values range from -1 to 1, with 1 representing full agreement between clustering and ground truth, 0 suggesting random labelling, and negative values showing less agreement than predicted by chance. A higher ARI score demonstrates that the model can properly group similar samples together and

[1]Details about the full names of the datasets used in the experiments can be found in Appendix C.
[2]The UCR dataset archive can be downloaded from here: https://www.cs.ucr.edu/%7Eeamonn/time_series_data_2018/

separate dissimilar ones; therefore, a higher ARI score corresponds to better ability of a model at extracting representations. ARI is calculated as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \cdot \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2} \cdot \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \cdot \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}, \tag{6}$$

where n is the total number of object instances, n_{ij} is the number of object instances in the i^{th} cluster of the clustering and the j^{th} group of the ground truth, and a_i and b_j are the number of rows and columns of the contingency table, respectively.

In addition to external evaluation, we conducted internal evaluation to examine the quality of clusters based on their structure (compactness and separation) without reference to ground truth labels. In this context, we used Calinski-Harabasz Index (CHI) as a metric. CHI is an internal cluster evaluation metric, with higher CHI scores indicating that the model constructs well-defined clusters of representations that preserve the rank order of similarities within the embedding space; therefore, a higher CHI score corresponds to better ability of a model at extracting representations. CHI is calculated as shown in Equation 7.

$$\text{CH Index} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \cdot \frac{N-k}{k-1}, \tag{7}$$

where $\text{Tr}(B_k)$ denotes the trace of the between-group dispersion matrix[3], $\text{Tr}(W_k)$ denotes the trace of the within-cluster dispersion matrix, N denotes the total number of samples in the dataset, and k denotes the number of clusters.

The evaluation process consists of comparing the clustering performance based on the following three scenarios: training the teacher model alone, training the student model alone, and training the student model with hints from the pre-trained teacher model within the TSRC framework. The training is performed on the training set, followed by extracting representations using the testing set. These representations are then clustered using the K-means clustering algorithm, with K being set as the number of classes for each dataset (calculated based on the labels provided with the dataset), and evaluated internally and externally using CHI and ARI metrics, respectively. The results of the three scenarios and both the internal and external cluster evaluations are then collated. A Friedman test followed by a Nemenyi post-hoc test are then performed for each evaluation metric to acquire final rankings and assess the statistical significance of the results for both internal and external cluster evaluations. The results of the tests are then visualised as Critical Difference Diagrams.

As demonstrated by Yue et al (2022), producing universal time series representations that are transferable between downstream tasks is a desirable property of time series representation learning models. To facilitate comparison with related time series classification methods (Middlehurst et al, 2024) and to evaluate the transferability of the representations produced by the models trained using TSRC, we performed classification using the baseline models and models trained as described earlier. Following

[3]The trace of a matrix is defined as the sum of elements on the main diagonal (given a square matrix A, the trace of A is defined as $\text{tr}(A) = \sum_{i=1}^{n} a_{ii}$).

Yue et al (2022), we used Suport Vector Machine (SVM) with an RBF kernel as the classification algorithm and accuracy as the evaluation metric.

Setting the Representation Length. The representation length is systematically computed for each dataset, based on the length of the input time series and a compression level (input hyperparameter). The calculation is performed as follows: Given the length L of a time series segment and a compression level C, the length of the representations S_{out} is derived as shown in Equation 8. The idea is to determine n such that 2^n is closest to the result of L/C; e.g., for $L = 100$ and $C = 2$, $S_{out} = 2^6 = 64$.

$$S_{out} = \begin{cases} 2^{\lfloor \log_2(L/C) \rfloor}, & \text{if } \left|\frac{L}{C} - 2^{\lfloor \log_2(L/C) \rfloor}\right| \leq \left|2^{\lceil \log_2(L/C) \rceil} - \frac{L}{C}\right| \\ 2^{\lceil \log_2(L/C) \rceil}, & \text{otherwise.} \end{cases} \tag{8}$$

We set the compression level to 2 in all our experiments to provide a fair comparison between the models across datasets. This minimises the effect of the length of time series samples on the evaluation process, because the models are evaluated based on their ability to extract representations of length 2^n closest to half of the input time series signal length. This approach of systematising representation length calculation also facilitates experimentation on additional datasets or models in any follow-up work.

4.3 Baseline Models

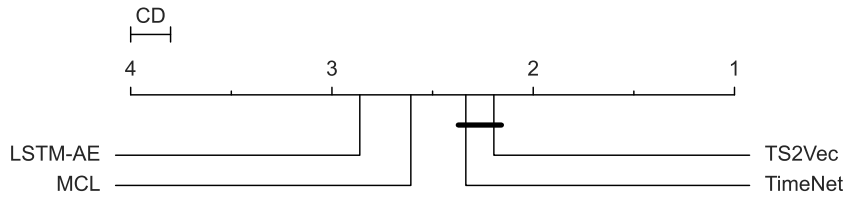
In our experiments, we considered multiple models from the literature as teacher and student models. The baselines were selected from contrastive- and reconstruction-based models (with publicly available source code[4].). We specifically selected these models based on their prior performance on popular benchmarks and insights reported in survey papers to be able to measure the impact of our framework, considering previously reported performance differences. The models considered in our experiments are as follows:

- Contrastive-based models:
  - TS2Vec (Yue et al, 2022): a contrastive-based model that utilises multiscale contextual information with different granularities to differentiate between samples.
  - Mixing Up Contrastive Learning (MCL) (Wickstrøm et al, 2022): a contrastive-based model that learns representations using noise injection.
- Reconstruction-based methods: each of the following methods is a reconstruction-based model that utilises a sequence auto-encoder (SAE) network based on the Seq2Seq model to transform variable-length time series into fixed-dimensional representations; the encoder and the decoder are trained jointly with the goal of minimising a reconstruction error.
  - TimeNet (Malhotra et al, 2017): a reconstruction-based model that utilises GRU layers.
  - LSTM Autoencoder (LSTM-AE) (Malhotra et al, 2016): a reconstruction-based model that utilises LSTM layers.

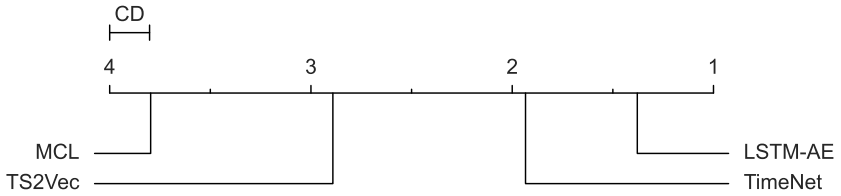
[4]Sources can be found in Appendix A

We conducted preliminary experiments to evaluate the performance of baseline models and used the results of these preliminary experiments to design subsequent experiments. Furthermore, we evaluated the performance of the baseline models based on the downstream task of time series clustering, as described in Section 4.2. Figures 4a and 4b show the Critical Difference (CD) diagrams for the baseline models for both external and internal cluster evaluations, respectively.

In our experiments, we intentionally selected student models that ranked lower compared to the teacher model based on external cluster evaluation (to ensure learning opportunity). The choice of an external cluster evaluation metric was motivated by the fact that these metrics measure the quality of embeddings in the embedding space by objectively assessing how closely similar samples are grouped together and how dissimilar samples are separated (based on ground truth labels). This aligns with the primary goal of the TSRC framework, which is to utilise both contrastive- and reconstruction-based methods to obtain the positioning of the embedding provided by contrastive learning while maintaining the interpretability provided by the decoder in reconstruction-based methods. While internal evaluation metrics are not used in designing the experiments, they are used to gauge the impact of the TSRC framework on the quality of the produced clusters, which in turn assesses whether it helps produce more well-defined clusters or not. We discuss the experiments and reasoning behind each of them in Section 5.1.



(a) A CD diagram that shows the ranking of the external cluster evaluation using Adjusted Rand Index (ARI).



(b) A CD diagram that shows the ranking of the internal cluster evaluation using Calinski-Harabasz Index (CHI).

Fig. 4: CD diagrams of clustering evaluation of the baseline models obtained by performing a Friedman test followed by a Nemenyi post-hoc test.

5 Results

In this section, the experiments are outlined and described alongside our thought process for both selecting the teacher and student models and designing the curriculum. The findings from the experiments are then listed and summarised, and research questions are listed and answered. The results can be replicated using the code of the framework and the experiments; it is open-source and available on GitHub [5].

5.1 Experiments

To study how the initial differences in the performances of the student and teacher models can influence the efficacy of training, we conducted a total of three experiments, where we covered various scenarios regarding the difference in performance between the teacher and the student model (e.g., significant performance differences and similar performance). The overall goal of these experiments was to evaluate the impact of the TSRC framework on the student model in the downstream tasks of time series clustering and classification. Specifically, we aimed to answer the following research questions:

[5] https://github.com/ADA-research/TSRC

1. Is the TSRC framework effective in allowing a reconstruction-based student model to learn from a contrastive-based teacher model?
2. How does the choice of the teacher and student models affect the performance of the TSRC framework?
3. How well do representations learned through the TSRC framework transfer to other downstream tasks, particularly time series classification?

The experiments are structured as follows:

- **Experiment 1:** Both teacher and student models have a high rank in external cluster evaluation (according to Figure 4a) and do not show a significant performance difference (strong-strong). Accordingly, we select TS2Vec (Yue et al, 2022) as the teacher model and TimeNet (Malhotra et al, 2017) as the student model. The curriculum is designed such that it gives less influence to the teacher model ($\mathcal{L}_{Hint}$) at the beginning and increases its impact later in the training process. This is accomplished by setting the increment rate $\tau = 1.5 > 1.0$ so that the value of the training coefficient λ_{train} starts low, giving more weight to the reconstruction loss, and then dynamically giving more weight to the hint loss, which is the teacher's influence (see Figures 5a and 3).
- **Experiment 2:** The teacher model has a high rank, the student has a lower rank in external cluster evaluation, and there is a significant performance difference between the two (strong-weak). In this experiment, we select TS2Vec (Yue et al, 2022) as teacher model and LSTM-AE (Malhotra et al, 2016) as student model. The curriculum is designed in a way to give more influence for the teacher model ($\mathcal{L}_{Hint}$) at the beginning, by setting the increment rate $\tau = 0.25 < 1.0$ (see Figures 5b and 3).
- **Experiment 3:** Both teacher and student models have low performance rankings in external cluster evaluation and show significant performance differences (weak-weak). In this experiment, we select MCL (Wickstrøm et al, 2022) as the teacher model and LSTM-AE (Malhotra et al, 2016) as the student model. The curriculum is designed to give a head start ($\psi_0 = 25$) to the student model to train without any influence from the teacher model at the beginning ($\lambda_{train} = 0 \Rightarrow \mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Reconstruction}}$); then the teacher influence is introduced in a balanced way, with linear increment achieved by setting the increment rate $\tau = 1.0$ (see Figures 5c and 3).

In these experiments, b, the final value of λ_{train}, is set based on both the performance of the teacher model and the performance difference between the teacher and student models. Specifically, when the teacher model is strong and the performance difference from the student model is not significant, b is set to 0.9, to allow some contribution from the student model at the end of training procedure (Experiment 1). When the teacher model is strong and significantly outperforms the student model, b is set to 1.0, to maximise the teacher's influence (Experiment 2). When the teacher model is not exceptionally strong but still better than the student model, b is set to 0.75, to reflect this moderate performance difference (Experiment 3).

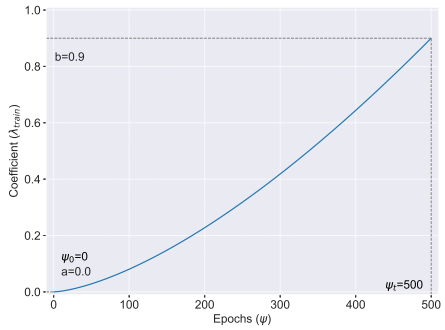
In all experiments, ψ_t (the total number of epochs) was set to 500 to ensure a consistent and fair comparison. In addition, as we seek to achieve the best interpretability possible, we aim to obtain the model with the encoder that produces presentations that

the decoder can reconstruct best; therefore, in our experiments, we set ($\lambda_{\text{validation}} = 0$) – when $\lambda_{\text{validation}}$ is set to 0, the validation favours models with the best reconstruction loss, i.e., a model with representations that can be best reconstructed by the student decoder (this is discussed in more detail in Section 3.2).

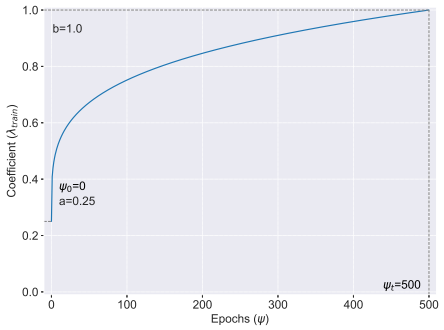
A summary of the hyperparameters (curricula) used in these experiments is presented in Table 1 and illustrated in Figure 5. The results for clustering and classification are summarised in Tables 2 and 3, and illustrated in Figures 6 and 7, respectively.

Table 1: Summary of the experiments.

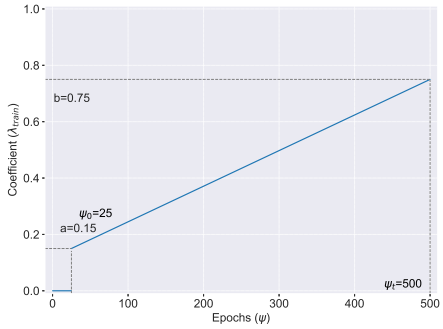
	Experiment 1 (strong-strong)	Experiment 2 (strong-weak)	Experiment 3 (weak-weak)
Models			
Teacher Model	TS2Vec	TS2Vec	MCL
Student Model	TimeNet	LSTM-AE	LSTM-AE
Curriculum hyperparameters			
a	0.0	0.25	0.15
b	0.9	1.0	0.75
ψ_0	0	0	25
ψ_t	500	500	500
τ	1.5	0.25	1.0
$\lambda_{validation}$	0.0	0.0	0.0



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Fig. 5: Change of the coefficient λ_{train} with respect to the hyperparameter settings (curriculum) and the epoch ψ for each experiment.

5.2 Discussion of Results

We conducted a total of three experiments (detailed in Section 5.1), in which we selected different combinations of teacher and student models, created a curriculum by adjusting the hyperparameters, and compared the performance of the student model when trained with and without the framework. Each experiment was run five times using different random seeds. To contextualise the performance of baseline models and models trained within the TSRC, we compared their clustering results to those obtained using features extracted directly from raw data. These features were extracted using a random sliding window approach[6], where mean values are computed over randomly selected intervals of the input time series. The number of sliding windows is determined using Equation 8. This feature extractor was run five times using the same random seeds as in other experiments.

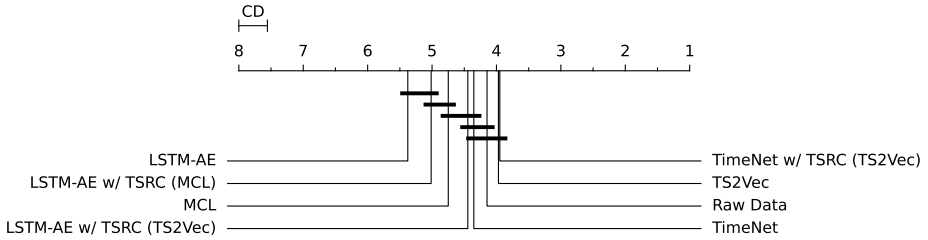
Q1. *Is the TSRC framework effective in allowing a reconstruction-based student model to learn from a contrastive-based teacher model?*

The results reported in Table 2 and Figure 6a show an overall improvement (on average 11.18%) in student model performance in terms of ranking in the external cluster evaluation based on experiments on 112 datasets when the student model was trained within the TSRC framework compared to performance when trained outside the framework. The results of the external cluster evaluation show that reconstruction-based methods could gain improvement in extracting clusterable representations, making them closer in performance to contrastive learning methods while keeping their enhanced interpretability (the trained decoder). In addition to improvements in external evaluation, methods trained within the TSRC gained a boost in their internal cluster evaluation, with an average improvement of 68.04% (see Figure 6b and Table 2), which means that the TSRC framework enhanced the ability of these models to extract more well-defined clusters. When looking at the clustering results acquired from the baseline models (shown in Figure 6), two conclusions can be drawn: (1) originally, baseline models produced worse representations than the features extracted from raw data (judged by internal clustering quality, see Figure 6b); however, (2) when trained using the TSRC framework, they produced significantly better representations than features extracted from raw data (see Figure 6b) without sacrificing external clustering performance (see Figure 6a).

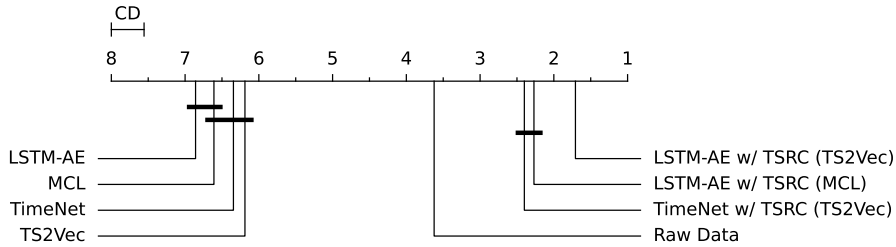
Q2. *How does the choice of the teacher and student models affect the performance of the TSRC framework?*

The degree of improvement varied in proportion to the difference in performance between student and teacher models. In Experiment 2, the performance of the student model improved significantly, compared to when it was trained outside the TSRC, whereas in Experiment 1, there was an improvement below statistical significance (see Figure 6a and Table 1). To further inspect the results of Experiment 3, which are ambiguous in Figure 6a – there is no significant difference between LSTM-AE w/ TSRC (MCL) and both MCL and LSTM-AE, but there is a significant difference between MCL and LSTM-AE –, we conducted a Wilcoxon signed-rank test on the

[6]We used the *RandomIntervalFeatureExtractor* implementation from "sktime" framework; implementation details are available here: https://www.sktime.net/en/v0.36.0/api_reference/auto_generated/sktime.transformations.panel.summarize.RandomIntervalFeatureExtractor.html.



(a) CD diagram of external cluster evaluation using Adjusted Rand Index (ARI) of the models considered in the experiments and the baselines obtained by performing a Friedman test followed by a Nemenyi post-hoc test.



(b) CD diagram of internal cluster evaluation using Calinski-Harabasz Index (CHI) of the models considered in the experiments and the baselines obtained by performing a Friedman test followed by a Nemenyi post-hoc test.

Fig. 6: Critical Difference (CD) diagrams of external and internal cluster evaluation of the results of the experiments and the baselines, using Adjusted Rand Index (ARI) and Calinski-Harabasz Index (CHI), respectively. The results were obtained by performing a Friedman test followed by a Nemenyi post-hoc test.

results of the student model and the student model trained using TSRC (i.e, LSTM-AE and LSTM-AE w/ TSRC (MCL), respectively). The result showed clearly that, at a standard significance level of 0.05, the observed difference in performance between the student and teacher models is significant ($p = 0.013$). These results can be explained by considering the initial ranking of the pool of baseline methods, as shown in Figure 4a. In Experiments 2 and 3, there was a significant difference between the performance values of the chosen student and teacher models, leading to a significant improvement in the performance of the student model. However, in Experiment 1, there was no significant difference between the chosen teacher and student models.

Q3. *How effectively do representations learned through the TSRC framework transfer to other downstream tasks, particularly time series classification?*

Classification was performed using the baseline models and models trained in the experiments described in Section 5.1. To ensure that transferability is properly

Table 2: The table shows the average rank of the external cluster evaluation using ARI and the internal cluster evaluation using CHI of the models considered in the experiments. The results were obtained by performing a Friedman test followed by a Nemenyi post-hoc test. It also shows the percentage rank improvement for the student models trained within the TSRC framework compared to the same student models trained individually. The naming convention used in this table for the TSRC framework is as follows: {student model name} w/ TSRC ({teacher model name})

External Cluster Evaluation (Adjusted Rand Index (ARI))		
	Average Rank	Average Rank Improvement (%)
Student models w/ TSRC framework		
TimeNet w/ TSRC (TS2Vec)	3.95	9.20% (Compared to TimeNet)
LSTM-AE w/ TSRC (TS2Vec)	4.44	17.47% (Compared to LSTM-AE)
LSTM-AE w/ TSRC (MCL)	5.01	6.88% (Compared to LSTM-AE)
Baselines		
TS2Vec	3.97	-
Raw Data	4.15	-
TimeNet	4.35	-
MCL	4.75	-
LSTM-AE	5.38	-
Internal Cluster Evaluation (Calinski-Harabasz Index (CHI))		
	Average Rank	Average Rank Improvement (%)
Student models w/ TSRC framework		
LSTM-AE w/ TSRC (TS2Vec)	1.71	75.07% (Compared to LSTM-AE)
LSTM-AE w/ TSRC (MCL)	2.77	66.91% (Compared to LSTM-AE)
TimeNet w/ TSRC (TS2Vec)	2.40	62.15% (Compared to TimeNet)
Baselines		
Raw Data	3.62	-
TS2Vec	6.19	-
TimeNet	6.34	-
MCL	6.61	-
LSTM-AE	6.86	-

evaluated, we did not re-evaluate the baseline methods, select new teacher-student combinations, or design new curricula based on classification performance; we used models trained using the curricula shown in Table 1. The classification results are presented in Table 3 and illustrated using the CD diagram in Figure 7. As can be observed from Table 3 and Figure 7, student models trained within the TSRC framework—using a suitable teacher model—extracted representations that achieved similar or better classification performance compared to the same models trained without TSRC. Specifically, as shown in Table 3, TimeNet w/ TSRC (TS2Vec) and LSTM-AE w/ TSRC (TS2Vec) were significantly better than TimeNet and LSTM-AE with average rank improvements of 18.06% and 9.99%, respectively. Although LSTM-AE w / TSRC (MCL) performs worse than MCL by 0.77% according to the results in Table 3, as

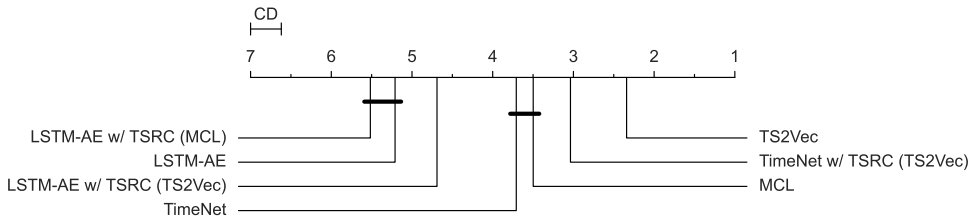


Fig. 7: Critical Difference (CD) diagrams of evaluations of the experiments and the baselines using Accuracy on the downstream task of time series classification. The results were obtained by performing a Friedman test followed by a Nemenyi post-hoc test.

Table 3: The table shows the average rank of the classficaiton evaluations using Accuracy of the models considered in the experiments. The results were obtained by performing a Friedman test followed by a Nemenyi post-hoc test. It also shows the percentage rank improvement for the student models trained within the TSRC framework compared to the same student models trained individually. The naming convention used in this table for the TSRC framework is as follows: {student model name} w/ TSRC ({teacher model name})

Classification Evaluation (Accuracy)		
	Average Rank	Average Rank Improvement (%)
Student models w/ TSRC framework		
TimeNet w/ TSRC (TS2Vec)	3.04	18.06% (Compared to TimeNet)
LSTM-AE w/ TSRC (TS2Vec)	4.69	9.99% (Compared to LSTM-AE)
LSTM-AE w/ TSRC (MCL)	2.06	-0.77% (Compared to LSTM-AE)
Baselines		
TS2Vec	2.34	-
MCL	3.50	-
TimeNet	3.71	-
LSTM-AE	5.21	-

Figure 7 shows, this performance difference is not statistically significant. These results suggest that TSRC not only enhances interpretability and the positions of samples in the embedding space but also improves the transferability of learned representations.

Based on these observations, we suggest the following general rule of thumb for using our new framework: given a properly designed curriculum, the teacher model should have a higher rank than the student model, with a significant difference in performance, to be able to effectively distil better knowledge from the teacher model.

The results presented in Table 2 suggest that the TSRC framework can assist reconstruction-based methods to extract better representations in terms of both internal and external cluster evaluation metrics without sacrificing the interpretability

provided by the decoder. This satisfies both of the desirable characteristics of the representations: representations are better positioned in the embedding space (improved external and internal cluster evaluation) and are interpretable (the models resulting from the framework are encoder-decoder models, which are inherently interpretable (Le Naour et al, 2023)). Furthermore, the results presented in Table 3 suggest that the TSRC improves the transferability of learned representations to other downstream tasks, which is also a desirable characteristic (Yue et al, 2022).

6 Conclusion

In this article, we have introduced the Time Series Representations Classroom (TSRC) framework and the accompanying loss function and curriculum, which we used to jointly train two models, a teacher and student model, to perform time series representation learning and evaluated it using the downstream tasks of time series clustering and classification. We demonstrated that this framework combines the benefits of contrastive- and reconstruction-based methods by combining their respective loss functions.

Our new framework can be used to improve the ability of reconstruction-based methods to extract more clusterable representations compared to when training is performed without the framework, while maintaining interpretability by making available a decoder after the training process. In our experiments, the Time Series Representations Classroom (TSRC) framework achieved a higher average ranking by at least 6.88% and up to 17.47% in external cluster evaluation, and by at least 62.15% and up to 75.07% in internal cluster evaluation. We have demonstrated that performance improvement is dependent on the difference in performance between the chosen teacher and student models and the designed curriculum. Our experiments provide insights into how these two models are chosen and how the curriculum is designed. Furthermore, our experiments demonstrated that the TSRC improved the transferability of the learned representations, where models trained using this framework achieved a higher average ranking in time series classification by 14.02% on average.

7 Limitations and Future Work

In our study, we have not considered adaptation to larger amounts of data (e.g., multiple datasets at once) of both the teacher and student models; in all our experiments, the teacher model was trained using one dataset at a time. In addition, robustness to anomalies and missing values was not taken into account when selecting the teacher and student models.

A promising avenue for future research consists of training a teacher model on multiple datasets, and then reusing it to teach one or more students; this could also include analysing and assessing the robustness of the teacher model and its impact on performance, potentially leading to pre-trained teacher models that can be used as a backbone for multiple runs of the TSRC framework (so far, the robustness of time series representation learning has not been much explored in the literature (Zhang et al, 2024a)).

Moreover, our experiments have been designed by manually selecting the teacher and student models and crafting a curriculum for each, which requires significant domain expertise and insight into the performance of both models. In order to democratise the usage of our framework, future work could include utilising Automated Machine Learning (AutoML) by building a search space that encompasses all hyperparameters, including choices for the student model, teacher model, and curriculum (Baratchi et al, 2024) to automatically select models and craft curricula based on previous performance; this could also consider model robustness when performing the selection. Additionally, future work could involve adding more choices for student and teacher models to the framework, allowing for more teacher-student model combinations in an attempt to realise a unified and standardised library, similar to ULTS (Meng et al, 2023).

Statements and Declarations

Acknowledgments. The authors would like to thank Anja Janković for reviewing and providing valuable feedback on the paper draft.
Part of the simulations and experiments were performed with computing resources granted by RWTH Aachen University under project "rwth1497".

Authors' contributions. **WS**: Conceptualisation, Data curation, Formal analysis, Investigation, Methodology, Software, Visualisation, Writing – original draft, Writing – review & editing. **MB**: Conceptualisation, Funding acquisition, Validation, Supervision, Writing – review & editing. **HH**: Conceptualisation, Funding acquisition, Validation, Supervision, Writing – review & editing.

Funding. Wadie Skaf and Holger Hoos have been partly funded by the Alexander von Humboldt Foundation (Alexander-von-Humboldt-Stiftung) through an Alexander von Humboldt Professorship awarded to Holger Hoos in 2022.
Mitra Baratchi has been partly funded by the research programme "Open Competition ENW" (partly financed by the Dutch Research Council (NWO)) under project "Physics-aware Spatio-temporal Machine Learning for Earth Observation Data" (Project No. OCENW.KLEIN.425).

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Data availability. The data used in this research is publicly available and can be downloaded from its official website: https://www.cs.ucr.edu/%7Eeamonn/time_series_data_2018/.

Code availability. The code to reproduce the results presented in this paper is open-source and publicly available on GitHub. The code can be accessed using the following link: https://github.com/ADA-research/TSRC.

References

Ay E, Devanne M, Weber J, et al (2022) A study of knowledge distillation in fully convolutional network for time series classification. In: Proceedings of International Joint Conference on Neural Networks, pp 1–8, https://doi.org/10.1109/IJCNN55064.2022.9892915

Banville H, Albuquerque I, Hyvärinen A, et al (2019) Self-supervised representation learning from electroencephalography signals. In: Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, pp 1–6, https://doi.org/10.1109/MLSP.2019.8918693

Baratchi M, Wang C, Limmer S, et al (2024) Automated machine learning: past, present and future. Artificial Intelligence Review 57(122). https://doi.org/10.1007/s10462-024-10726-1

Bengio Y, Louradour J, Collobert R, et al (2009) Curriculum learning. In: Proceedings of the International Conference on Machine Learning, p 41–48, https://doi.org/10.1145/1553374.1553380

Chen J, Zeng GQ, Zhou W, et al (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Conversion and Management 165:681–695. https://doi.org/10.1016/j.enconman.2018.03.098

Dau HA, Bagnall A, Kamgar K, et al (2019) The UCR time series archive. IEEE/CAA Journal of Automatica Sinica 6(6):1293–1305. https://doi.org/10.1109/JAS.2019.1911747

De Gooijer JG, Hyndman RJ (2006) 25 years of time series forecasting. International Journal of Forecasting 22(3):443–473. https://doi.org/10.1016/j.ijforecast.2006.01.001

Eldele E, Ragab M, Chen Z, et al (2021) Time-series representation learning via temporal and contextual contrasting. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp 2352–2359, https://doi.org/10.24963/ijcai.2021/324

Foumani NM, Miller L, Tan CW, et al (2024a) Deep learning for time series classification and extrinsic regression: A current survey. ACM Computing Surveys 56(9). https://doi.org/10.1145/3649448

Foumani NM, Tan CW, Webb GI, et al (2024b) Series2Vec: Similarity-based self-supervised representation learning for time series classification. Data Mining and Knowledge Discovery 38(4):2520–2544. https://doi.org/10.1007/s10618-024-01043-w

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, http://www.deeplearningbook.org

Habib G, Saleem TJ, Lall B (2023) Knowledge distillation in vision transformers: A critical review. URL https://arxiv.org/abs/2302.02108v2

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop

Hyvarinen A, Morioka H (2016) Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In: Proceedings of Advances in Neural Information Processing Systems

Ismail-Fawaz A, Devanne M, Berretti S, et al (2024) Finding foundation models for time series classification with a pretext task. In: In Proccedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 123–135, https://doi.org/10.1007/978-981-97-2650-9_10

Ismail Fawaz H, Forestier G, Weber J, et al (2019) Deep learning for time series classification: a review. Data Mining Knowledge Discovery 33(4):917–963. https://doi.org/10.1007/s10618-019-00619-1

Le Naour E, Agoua G, Baskiotis N, et al (2023) Interpretable time series neural representation for classification purposes. In: Proceedings of the International Conference on Data Science and Advanced Analytics, pp 1–10, https://doi.org/10.1109/DSAA60987.2023.10302534

Li D, Chen D, Jin B, et al (2019) MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In: Proceedings of International Conference on Artificial Neural Networks, pp 703–716, https://doi.org/10.1007/978-3-030-30490-4_56

Liu F, Zhang T, Zhang C, et al (2023) A review of the evaluation system for curriculum learning. Electronics 12(7). https://doi.org/10.3390/electronics12071676

Längkvist M, Karlsson L, Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters 42:11–24. https://doi.org/10.1016/j.patrec.2014.01.008

Ma Q, Zheng J, Li S, et al (2019) Learning representations for time series clustering. In: Proceedings of Advances in Neural Information Processing Systems

Ma Q, Chen C, Li S, et al (2021) Learning representations for incomplete time series clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, https://doi.org/10.1609/aaai.v35i10.17070

Malhotra P, Ramakrishnan A, Anand G, et al (2016) LSTM-based encoder-decoder for multi-sensor anomaly detection. URL https://arxiv.org/abs/1607.00148v2

Malhotra P, TV V, Vig L, et al (2017) TimeNet: Pre-trained deep recurrent neural network for time series classification. In: Proceedings of the European Symposium on Artificial Neural Networks

Meng Q, Qian H, Liu Y, et al (2023) Unsupervised representation learning for time series: A review. URL https://arxiv.org/abs/2308.01578v1

Middlehurst M, Schäfer P, Bagnall A (2024) Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Mining and Knowledge Discovery 38(4):1958–2031. https://doi.org/10.1007/s10618-024-01022-1

Mohamed A, Lee Hy, Borgholt L, et al (2022) Self-supervised speech representation learning: A review. IEEE Journal of Selected Topics in Signal Processing 16(6):1179–1210. https://doi.org/10.1109/JSTSP.2022.3207050

Qiao Z, Hu M, Jiang X, et al (2023) Class-incremental learning on multivariate time series via shape-aligned temporal distillation. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp 1–5, https://doi.org/10.1109/ICASSP49357.2023.10094960

Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing 90:106181. https://doi.org/10.1016/j.asoc.2020.106181

Skaf W, Horváth T (2022) Denoising architecture for unsupervised anomaly detection in time-series. In: Proceedings of European Conference on Advances in Databases and Information Systems, pp 178–187, https://doi.org/10.1007/978-3-031-15743-1_17

Skaf W, Tosayeva A, Várkonyi DT (2023) Towards automatic forecasting: Evaluation of time-series forecasting models for chickenpox cases estimation in Hungary. In: Proceedings of Intelligent Systems Design and Applications, pp 1–10, https://doi.org/10.1007/978-3-031-35507-3_1

Song W, Liu L, Liu M, et al (2020) Representation learning with deconvolution for multivariate time series classification and visualization. In: Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, pp 310–326, https://doi.org/10.1007/978-981-15-7981-3_22

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, p 3104–3112

Wickstrøm K, Kampffmeyer M, Øyvind Mikalsen K, et al (2022) Mixing up contrastive learning: Self-supervised representation learning for time series. Pattern Recognition Letters 155:54–61. https://doi.org/10.1016/j.patrec.2022.02.007

Yoon J, Jarrett D, van der Schaar M (2019) Time-series generative adversarial networks. In: Proceddings of Advances in Neural Information Processing Systems

Yue Z, Wang Y, Duan J, et al (2022) TS2Vec: Towards universal representation of time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 8980–8987, https://doi.org/10.1609/aaai.v36i8.20881

Zerveas G, Jayaraman S, Patel D, et al (2021) A transformer-based framework for multivariate time series representation learning. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, p 2114–2124, https://doi.org/10.1145/3447548.3467401

Zhang K, Wen Q, Zhang C, et al (2024a) Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. IEEE Transactions on Pattern Analysis and Machine Intelligence pp 1–20. https://doi.org/10.1109/TPAMI.2024.3387317

Zhang Q, Wu J, Zhang P, et al (2019) Salient subsequence learning for time series clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(9):2193–2207. https://doi.org/10.1109/TPAMI.2018.2847699

Zhang W, Yang L, Geng S, et al (2024b) Self-supervised time series representation learning via cross reconstruction transformer. IEEE Transactions on Neural Networks and Learning Systems 35(11):16129–16138. https://doi.org/10.1109/TNNLS.2023.3292066

Zhao H, Sun X, Dong J, et al (2021) Knowledge distillation via instance-level sequence learning. Knowledge-Based Systems 233:107519. https://doi.org/10.1016/j.knosys.2021.107519

Appendix A Code Sources for The Baseline Models

In Table A1, we list the code sources we used to implement the baseline models discussed in Section 4.3.

Model	Source
TS2Vec (Yue et al, 2022)	https://github.com/zhihanyue/ts2vec
TimeNet (Malhotra et al, 2017)	https://github.com/mqwfrog/ULTS/tree/main/models/TimeNet
MCL (Wickstrøm et al, 2022)	https://github.com/Wickstrom/MixupContrastiveLearning
LSTM-AE (Malhotra et al, 2016)	https://github.com/PyLink88/Recurrent-Autoencoder

Table A1: Code source for baseline models considered in our study.

Appendix B Note on the Classification Results (Transferability)

Although we have demonstrated that representations extracted using TSRC in an unsupervised fashion are transferable to downstream tasks that are inherently supervised (classification), in case enough labels for training and validating are available, it might be better to use supervised training methods for representation learning. Using TSRC only for downstream tasks that are inherently based on supervised learning makes selecting the teacher model and designing a curriculum a more difficult task than using supervised learning-based methods (other than TSRC).

Appendix C Detailed Results

In Tables C2 and C3, we list the full results of the external and the internal cluster evaluations conducted in our experiments, discussed in Section 5, for every dataset we considered. In Table C4 we list the detailed results for the classification downstream task. Every experiment was run 5 times; the random seeds used in the 5 runs are: 1, 3, 5, 7, and 9.

Table C2: Detailed results of the external cluster evaluation from the experiments and the evaluation of the baselines. The number denotes (mean ± standard deviation) of the ARI metric from 5 runs of each experiment. The rows represent the datasets and the columns represent the model, either a baseline or a model trained within the TSRC framework. The naming convention used in this table for the TSRC framework is as follows: {student model name} w/ TSRC ({teacher model name}).

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
ACSF1	0.09 ± 0.00	0.22 ± 0.01	0.18 ± 0.03	0.26 ± 0.01	0.22 ± 0.02	0.20 ± 0.04	0.24 ± 0.01	0.19 ± 0.01
Adiac	0.14 ± 0.01	0.12 ± 0.03	0.25 ± 0.01	0.30 ± 0.02	0.15 ± 0.01	0.23 ± 0.04	0.18 ± 0.03	0.22 ± 0.03
ArrowHead	0.09 ± 0.01	0.05 ± 0.01	-0.01 ± 0.00	0.09 ± 0.04	0.05 ± 0.02	0.11 ± 0.03	0.09 ± 0.03	0.03 ± 0.03
BME	0.35 ± 0.01	0.22 ± 0.07	-0.00 ± 0.00	0.25 ± 0.07	0.37 ± 0.18	0.33 ± 0.13	0.21 ± 0.10	0.31 ± 0.11

Continued on next page

Table C2 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
Beef	0.06 ± 0.01	0.04 ± 0.00	0.08 ± 0.03	0.18 ± 0.07	0.08 ± 0.04	0.11 ± 0.02	0.04 ± 0.00	0.11 ± 0.05
BeetleFly	0.12 ± 0.00	0.12 ± 0.00	0.20 ± 0.04	0.04 ± 0.05	0.10 ± 0.05	0.04 ± 0.05	0.12 ± 0.00	0.11 ± 0.03
BirdChicken	-0.05 ± 0.00	-0.01 ± 0.00	-0.03 ± 0.02	-0.01 ± 0.03	-0.01 ± 0.00	-0.03 ± 0.02	-0.01 ± 0.00	-0.00 ± 0.01
CBF	0.25 ± 0.04	0.26 ± 0.13	0.63 ± 0.05	0.30 ± 0.13	0.18 ± 0.04	0.28 ± 0.13	0.72 ± 0.09	0.81 ± 0.07
Car	0.12 ± 0.01	0.02 ± 0.00	0.06 ± 0.01	0.15 ± 0.04	0.08 ± 0.02	0.13 ± 0.04	0.06 ± 0.03	0.08 ± 0.02
Chinatown	0.08 ± 0.01	0.71 ± 0.28	0.08 ± 0.02	0.03 ± 0.03	0.56 ± 0.30	0.60 ± 0.28	0.02 ± 0.02	0.03 ± 0.05
ChlorineConcentration	-0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00
CinCECGTorso	0.13 ± 0.00	0.03 ± 0.01	0.08 ± 0.02	0.07 ± 0.02	0.10 ± 0.01	0.03 ± 0.01	0.05 ± 0.03	0.04 ± 0.01
Coffee	0.01 ± 0.02	0.05 ± 0.00	-0.02 ± 0.00	-0.00 ± 0.01	0.05 ± 0.00	0.18 ± 0.16	0.05 ± 0.01	0.05 ± 0.03
Computers	-0.00 ± 0.00	-0.00 ± 0.00	0.02 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	-0.00 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
CricketX	0.07 ± 0.01	0.03 ± 0.00	0.10 ± 0.00	0.07 ± 0.01	0.06 ± 0.01	0.05 ± 0.01	0.08 ± 0.01	0.08 ± 0.01
CricketY	0.09 ± 0.01	0.07 ± 0.01	0.12 ± 0.02	0.08 ± 0.01	0.08 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.13 ± 0.01
CricketZ	0.07 ± 0.01	0.03 ± 0.00	0.13 ± 0.01	0.09 ± 0.02	0.04 ± 0.01	0.05 ± 0.01	0.07 ± 0.03	0.08 ± 0.01
Crop	0.26 ± 0.02	0.25 ± 0.02	0.27 ± 0.01	0.27 ± 0.01	0.26 ± 0.04	0.24 ± 0.04	0.25 ± 0.02	0.24 ± 0.01
DiatomSizeReduction	0.48 ± 0.01	0.43 ± 0.15	0.03 ± 0.00	0.11 ± 0.16	0.82 ± 0.02	0.77 ± 0.07	0.83 ± 0.05	0.86 ± 0.01
DistalPhalanxOutlineAgeGroup	0.45 ± 0.02	0.36 ± 0.11	0.39 ± 0.01	0.39 ± 0.03	0.44 ± 0.01	0.33 ± 0.11	0.44 ± 0.02	0.44 ± 0.01
DistalPhalanxOutlineCorrect	-0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00
DistalPhalanxTW	0.75 ± 0.01	0.69 ± 0.11	0.73 ± 0.01	0.34 ± 0.07	0.68 ± 0.13	0.29 ± 0.02	0.74 ± 0.01	0.74 ± 0.01
ECG200	0.25 ± 0.02	0.19 ± 0.06	0.16 ± 0.09	0.22 ± 0.17	0.23 ± 0.06	0.29 ± 0.04	0.24 ± 0.04	0.25 ± 0.03
ECG5000	0.62 ± 0.05	0.44 ± 0.03	0.46 ± 0.01	0.28 ± 0.04	0.48 ± 0.06	0.36 ± 0.04	0.46 ± 0.02	0.48 ± 0.02
ECGFiveDays	0.08 ± 0.02	0.24 ± 0.05	0.01 ± 0.01	0.10 ± 0.07	0.15 ± 0.16	0.38 ± 0.22	0.00 ± 0.01	0.00 ± 0.01
EOGHorizontalSignal	0.17 ± 0.02	0.10 ± 0.00	0.28 ± 0.02	0.19 ± 0.04	0.15 ± 0.01	0.10 ± 0.00	0.10 ± 0.00	0.14 ± 0.04
EOGVerticalSignal	0.07 ± 0.01	0.04 ± 0.00	0.05 ± 0.00	0.08 ± 0.01	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00
Earthquakes	-0.01 ± 0.01	0.24 ± 0.01	-0.06 ± 0.01	-0.00 ± 0.00	0.25 ± 0.00	0.13 ± 0.08	0.23 ± 0.00	0.24 ± 0.01
ElectricDevices	0.17 ± 0.03	0.13 ± 0.02	0.25 ± 0.02	0.04 ± 0.00	0.14 ± 0.02	0.13 ± 0.04	0.25 ± 0.02	0.25 ± 0.02
EthanolLevel	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
FaceAll	0.24 ± 0.04	0.13 ± 0.07	0.16 ± 0.02	0.18 ± 0.02	0.08 ± 0.03	0.16 ± 0.08	0.16 ± 0.02	0.22 ± 0.04
FaceFour	0.39 ± 0.08	0.21 ± 0.01	0.07 ± 0.05	0.34 ± 0.14	0.19 ± 0.06	0.20 ± 0.08	0.20 ± 0.05	0.27 ± 0.15
FacesUCR	0.15 ± 0.03	0.16 ± 0.02	0.19 ± 0.02	0.17 ± 0.03	0.20 ± 0.03	0.09 ± 0.02	0.24 ± 0.01	0.27 ± 0.03
FiftyWords	0.29 ± 0.02	0.11 ± 0.01	0.30 ± 0.02	0.23 ± 0.01	0.24 ± 0.04	0.11 ± 0.04	0.14 ± 0.03	0.20 ± 0.04
Fish	0.09 ± 0.01	0.07 ± 0.01	0.08 ± 0.02	0.25 ± 0.05	0.09 ± 0.01	0.11 ± 0.01	0.08 ± 0.01	0.11 ± 0.02
FordA	-0.00 ± 0.00	-0.00 ± 0.00	0.01 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00
FordB	0.00 ± 0.00	-0.00 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	0.00 ± 0.00
FreezerRegularTrain	0.28 ± 0.01	0.20 ± 0.07	0.24 ± 0.01	0.05 ± 0.03	0.26 ± 0.01	0.09 ± 0.08	0.23 ± 0.03	0.24 ± 0.01
FreezerSmallTrain	0.32 ± 0.00	0.24 ± 0.08	0.21 ± 0.01	0.05 ± 0.04	0.12 ± 0.10	0.27 ± 0.06	0.23 ± 0.01	0.23 ± 0.01
Fungi	0.52 ± 0.02	0.08 ± 0.02	0.54 ± 0.05	0.50 ± 0.08	0.31 ± 0.08	0.36 ± 0.11	0.13 ± 0.04	0.27 ± 0.08
GunPoint	-0.00 ± 0.01	0.01 ± 0.01	0.05 ± 0.04	0.02 ± 0.03	0.01 ± 0.01	0.02 ± 0.03	0.00 ± 0.02	-0.01 ± 0.00
GunPointAgeSpan	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	0.06 ± 0.04	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00	-0.00 ± 0.00
GunPointMaleVersusFemale	0.21 ± 0.00	0.21 ± 0.00	0.21 ± 0.00	0.12 ± 0.11	0.21 ± 0.00	0.13 ± 0.10	0.21 ± 0.00	0.17 ± 0.08
GunPointOldVersusYoung	0.28 ± 0.00	0.29 ± 0.02	0.28 ± 0.00	0.23 ± 0.09	0.56 ± 0.34	0.28 ± 0.00	0.28 ± 0.00	0.28 ± 0.00
Ham	-0.01 ± 0.00	-0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.01 ± 0.00	0.03 ± 0.03
Haptics	0.06 ± 0.00	0.03 ± 0.00	0.07 ± 0.02	0.04 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.02
Herring	-0.01 ± 0.00	-0.01 ± 0.00	-0.02 ± 0.00	-0.00 ± 0.02	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.01
HouseTwenty	0.05 ± 0.00	0.04 ± 0.00	0.00 ± 0.01	0.08 ± 0.02	0.04 ± 0.00	0.03 ± 0.02	0.03 ± 0.00	0.03 ± 0.01
InlineSkate	0.03 ± 0.00	0.03 ± 0.01	0.02 ± 0.00	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.00	0.03 ± 0.01
InsectEPGRegularTrain	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
InsectEPGSmallTrain	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.73 ± 0.22	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
InsectWingbeatSound	0.24 ± 0.03	0.21 ± 0.06	0.32 ± 0.02	0.26 ± 0.02	0.29 ± 0.04	0.17 ± 0.05	0.27 ± 0.02	0.29 ± 0.03
ItalyPowerDemand	0.00 ± 0.00	0.02 ± 0.01	0.05 ± 0.05	0.08 ± 0.06	0.00 ± 0.00	0.05 ± 0.08	0.08 ± 0.05	0.05 ± 0.04
LargeKitchenAppliances	0.03 ± 0.01	-0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.03 ± 0.02	0.03 ± 0.01	0.01 ± 0.02
Lightning2	0.10 ± 0.03	0.01 ± 0.01	-0.00 ± 0.00	0.11 ± 0.10	0.01 ± 0.02	0.01 ± 0.05	0.01 ± 0.01	0.00 ± 0.01
Lightning7	0.31 ± 0.01	0.10 ± 0.01	0.11 ± 0.02	0.27 ± 0.06	0.19 ± 0.02	0.13 ± 0.01	0.16 ± 0.07	0.27 ± 0.03
Mallat	0.70 ± 0.05	0.04 ± 0.01	0.83 ± 0.04	0.30 ± 0.03	0.41 ± 0.05	0.06 ± 0.02	0.31 ± 0.36	0.31 ± 0.29
Meat	0.44 ± 0.04	0.05 ± 0.02	-0.01 ± 0.02	0.43 ± 0.06	0.34 ± 0.11	0.20 ± 0.04	0.42 ± 0.20	0.38 ± 0.18
MedicalImages	0.06 ± 0.01	0.05 ± 0.02	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.05 ± 0.01	0.03 ± 0.01	0.04 ± 0.01
MiddlePhalanxOutlineAgeGroup	0.45 ± 0.00	0.44 ± 0.03	0.47 ± 0.01	0.36 ± 0.05	0.44 ± 0.01	0.42 ± 0.02	0.44 ± 0.03	0.44 ± 0.02
MiddlePhalanxOutlineCorrect	-0.00 ± 0.00	-0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.01	-0.00 ± 0.01	-0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.01
MiddlePhalanxTW	0.52 ± 0.09	0.52 ± 0.09	0.54 ± 0.01	0.29 ± 0.11	0.46 ± 0.11	0.32 ± 0.04	0.51 ± 0.09	0.57 ± 0.01
MixedShapesRegularTrain	0.47 ± 0.01	0.34 ± 0.15	0.50 ± 0.01	0.22 ± 0.05	0.36 ± 0.08	0.05 ± 0.01	0.24 ± 0.11	0.29 ± 0.15
MixedShapesSmallTrain	0.47 ± 0.01	0.15 ± 0.11	0.49 ± 0.02	0.27 ± 0.06	0.34 ± 0.06	0.06 ± 0.04	0.11 ± 0.08	0.20 ± 0.11
MoteStrain	0.15 ± 0.11	0.30 ± 0.15	0.04 ± 0.01	0.12 ± 0.12	0.34 ± 0.05	0.18 ± 0.08	0.12 ± 0.14	0.31 ± 0.07
NonInvasiveFetalECGThorax1	0.27 ± 0.02	0.16 ± 0.05	0.38 ± 0.01	0.32 ± 0.02	0.36 ± 0.05	0.14 ± 0.03	0.30 ± 0.02	0.35 ± 0.02
NonInvasiveFetalECGThorax2	0.27 ± 0.01	0.38 ± 0.03	0.37 ± 0.02	0.33 ± 0.01	0.45 ± 0.02	0.20 ± 0.04	0.35 ± 0.02	0.38 ± 0.02
OSULeaf	0.15 ± 0.01	0.06 ± 0.01	0.21 ± 0.04	0.08 ± 0.02	0.10 ± 0.01	0.12 ± 0.02	0.07 ± 0.04	0.08 ± 0.02

Continued on next page

Table C2 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
OliveOil	0.81 ± 0.06	0.09 ± 0.00	0.14 ± 0.02	0.75 ± 0.10	0.18 ± 0.12	0.12 ± 0.02	0.09 ± 0.00	0.31 ± 0.11
PhalangesOutlinesCorrect	0.01 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	0.01 ± 0.00
Phoneme	-0.01 ± 0.00	0.02 ± 0.00	0.06 ± 0.01	0.00 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.01	0.02 ± 0.01
PigAirwayPressure	0.05 ± 0.01	0.03 ± 0.00	0.18 ± 0.02	0.05 ± 0.01	0.03 ± 0.00	0.04 ± 0.01	0.05 ± 0.01	0.04 ± 0.01
PigArtPressure	0.08 ± 0.01	0.01 ± 0.01	0.44 ± 0.02	0.14 ± 0.01	0.00 ± 0.01	0.03 ± 0.01	0.03 ± 0.02	0.03 ± 0.03
PigCVP	0.06 ± 0.00	0.01 ± 0.01	0.12 ± 0.01	0.02 ± 0.01	0.05 ± 0.02	0.02 ± 0.02	0.03 ± 0.03	0.04 ± 0.02
Plane	0.75 ± 0.02	0.53 ± 0.01	0.89 ± 0.05	0.66 ± 0.07	0.43 ± 0.05	0.80 ± 0.05	0.53 ± 0.08	0.68 ± 0.09
PowerCons	0.56 ± 0.09	0.01 ± 0.00	0.41 ± 0.09	0.03 ± 0.04	0.01 ± 0.00	0.41 ± 0.27	0.55 ± 0.04	0.58 ± 0.04
ProximalPhalanxOutlineAgeGroup	0.53 ± 0.01	0.49 ± 0.07	0.48 ± 0.02	0.23 ± 0.09	0.49 ± 0.07	0.46 ± 0.13	0.52 ± 0.01	0.52 ± 0.01
ProximalPhalanxOutlineCorrect	0.07 ± 0.00	0.07 ± 0.01	0.00 ± 0.01	0.02 ± 0.01	0.06 ± 0.01	0.05 ± 0.02	-0.00 ± 0.01	0.01 ± 0.02
ProximalPhalanxTW	0.42 ± 0.02	0.38 ± 0.05	0.37 ± 0.03	0.28 ± 0.03	0.47 ± 0.05	0.30 ± 0.02	0.49 ± 0.09	0.46 ± 0.07
RefrigerationDevices	0.00 ± 0.00	0.01 ± 0.00	0.06 ± 0.01	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.00
Rock	0.06 ± 0.00	0.07 ± 0.01	0.07 ± 0.00	0.20 ± 0.05	0.06 ± 0.00	0.07 ± 0.01	0.08 ± 0.01	0.12 ± 0.04
ScreenType	0.01 ± 0.00	0.01 ± 0.01	0.03 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01
SemgHandGenderCh2	-0.02 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.01	-0.02 ± 0.01	-0.01 ± 0.00	-0.01 ± 0.01	0.01 ± 0.02	-0.01 ± 0.01
SemgHandMovementCh2	0.14 ± 0.00	0.09 ± 0.00	0.18 ± 0.01	0.12 ± 0.00	0.12 ± 0.01	0.12 ± 0.02	0.09 ± 0.05	0.12 ± 0.02
SemgHandSubjectCh2	0.08 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.10 ± 0.01	0.10 ± 0.02	0.08 ± 0.01	0.07 ± 0.01	0.07 ± 0.02
ShapeletSim	-0.01 ± 0.00	-0.01 ± 0.00	1.00 ± 0.00	0.00 ± 0.01	-0.01 ± 0.00	0.00 ± 0.01	-0.01 ± 0.00	-0.01 ± 0.00
ShapesAll	0.30 ± 0.00	0.22 ± 0.05	0.43 ± 0.01	0.36 ± 0.01	0.31 ± 0.02	0.14 ± 0.03	0.23 ± 0.02	0.33 ± 0.01
SmallKitchenAppliances	0.00 ± 0.00	0.09 ± 0.05	0.02 ± 0.01	0.00 ± 0.00	0.01 ± 0.01	0.15 ± 0.07	0.00 ± 0.00	0.00 ± 0.00
SmoothSubspace	0.32 ± 0.05	0.22 ± 0.11	0.13 ± 0.04	0.13 ± 0.05	0.43 ± 0.06	0.27 ± 0.11	0.48 ± 0.07	0.44 ± 0.12
SonyAIBORobotSurface1	0.04 ± 0.05	0.49 ± 0.16	0.01 ± 0.01	0.05 ± 0.05	0.31 ± 0.23	0.27 ± 0.22	0.02 ± 0.02	0.04 ± 0.07
SonyAIBORobotSurface2	0.13 ± 0.07	0.22 ± 0.04	0.09 ± 0.06	0.25 ± 0.12	0.26 ± 0.02	0.19 ± 0.03	0.19 ± 0.05	0.19 ± 0.03
StarLightCurves	0.52 ± 0.00	0.53 ± 0.04	0.48 ± 0.01	0.28 ± 0.21	0.55 ± 0.01	0.36 ± 0.18	0.52 ± 0.02	0.53 ± 0.06
Strawberry	-0.03 ± 0.00	-0.01 ± 0.01	-0.00 ± 0.00	-0.01 ± 0.02	0.01 ± 0.02	0.12 ± 0.06	0.00 ± 0.01	-0.00 ± 0.00
SwedishLeaf	0.26 ± 0.02	0.25 ± 0.03	0.39 ± 0.04	0.28 ± 0.03	0.30 ± 0.05	0.31 ± 0.02	0.39 ± 0.03	0.37 ± 0.05
Symbols	0.68 ± 0.01	0.49 ± 0.15	0.87 ± 0.02	0.62 ± 0.03	0.65 ± 0.02	0.60 ± 0.06	0.72 ± 0.04	0.84 ± 0.01
SyntheticControl	0.53 ± 0.04	0.56 ± 0.04	0.93 ± 0.02	0.33 ± 0.06	0.54 ± 0.04	0.43 ± 0.03	0.59 ± 0.08	0.59 ± 0.04
ToeSegmentation1	-0.00 ± 0.00	-0.00 ± 0.00	0.16 ± 0.01	-0.00 ± 0.00	0.00 ± 0.01	-0.00 ± 0.01	-0.00 ± 0.00	-0.00 ± 0.00
ToeSegmentation2	0.01 ± 0.01	0.02 ± 0.00	0.11 ± 0.02	0.01 ± 0.02	0.03 ± 0.00	-0.01 ± 0.01	0.04 ± 0.01	0.05 ± 0.04
Trace	0.37 ± 0.00	0.24 ± 0.06	0.43 ± 0.02	0.27 ± 0.12	0.58 ± 0.06	0.62 ± 0.03	0.37 ± 0.22	0.55 ± 0.08
TwoLeadECG	0.01 ± 0.01	0.02 ± 0.02	-0.00 ± 0.00	0.02 ± 0.02	0.01 ± 0.02	0.02 ± 0.03	0.00 ± 0.00	-0.00 ± 0.00
TwoPatterns	0.05 ± 0.01	-0.00 ± 0.00	0.62 ± 0.07	0.02 ± 0.01	0.00 ± 0.00	0.04 ± 0.01	0.65 ± 0.05	0.64 ± 0.10
UMD	0.18 ± 0.04	0.01 ± 0.02	0.00 ± 0.00	0.14 ± 0.03	0.11 ± 0.11	0.14 ± 0.01	0.11 ± 0.05	0.15 ± 0.03
UWaveGestureLibraryAll	0.44 ± 0.01	0.13 ± 0.14	0.27 ± 0.03	0.35 ± 0.02	0.25 ± 0.17	0.10 ± 0.03	0.23 ± 0.18	0.23 ± 0.13
UWaveGestureLibraryX	0.30 ± 0.01	0.17 ± 0.04	0.29 ± 0.04	0.26 ± 0.03	0.22 ± 0.03	0.18 ± 0.04	0.30 ± 0.03	0.29 ± 0.03
UWaveGestureLibraryY	0.29 ± 0.00	0.27 ± 0.03	0.18 ± 0.01	0.25 ± 0.03	0.28 ± 0.01	0.27 ± 0.03	0.18 ± 0.02	0.19 ± 0.02
UWaveGestureLibraryZ	0.28 ± 0.01	0.25 ± 0.01	0.30 ± 0.03	0.24 ± 0.01	0.29 ± 0.01	0.19 ± 0.04	0.30 ± 0.03	0.31 ± 0.02
Wafer	-0.01 ± 0.00	-0.01 ± 0.00	0.02 ± 0.04	0.01 ± 0.02	-0.01 ± 0.00	-0.00 ± 0.00	0.02 ± 0.03	0.00 ± 0.00
Wine	-0.00 ± 0.01	-0.01 ± 0.00	-0.01 ± 0.01	0.04 ± 0.05	0.01 ± 0.02	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.02
WordSynonyms	0.16 ± 0.02	0.06 ± 0.01	0.20 ± 0.02	0.14 ± 0.01	0.16 ± 0.01	0.09 ± 0.02	0.12 ± 0.02	0.20 ± 0.02
Worms	0.03 ± 0.00	0.04 ± 0.00	0.05 ± 0.01	0.00 ± 0.01	0.06 ± 0.01	0.03 ± 0.02	0.06 ± 0.02	0.05 ± 0.01
WormsTwoClass	0.00 ± 0.00	-0.00 ± 0.00	0.05 ± 0.02	-0.00 ± 0.00	0.00 ± 0.01	-0.01 ± 0.00	-0.00 ± 0.00	0.00 ± 0.01
Yoga	-0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

Table C3: Detailed results of the internal cluster evaluation from the experiments and the evaluation of the baselines. The number denotes (mean ± standard deviation) of the CHI metric from 5 runs of each experiment. The rows represent the datasets and the columns represent the model, either a baseline or a model trained within the TSRC framework. The naming convention used in this table for the TSRC framework is as follows: {student model name} w/ TSRC ({teacher model name}).

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
ACSF1	10.42 ± 0.88	124.22 ± 27.73	164.76 ± 33.72	67.70 ± 13.01	106.62 ± 23.38	417.49 ± 413.78	131.00 ± 10.24	87.38 ± 14.33
Adiac	131.97 ± 9.10	549.68 ± 322.67	106.86 ± 19.94	93.71 ± 9.83	410.73 ± 136.96	110.43 ± 29.15	350.71 ± 103.66	187.46 ± 17.90

Continued on next page

Table C3 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
ArrowHead	25.07 ± 1.53	119.86 ± 32.01	63.58 ± 9.14	33.99 ± 6.30	70.94 ± 24.64	56.73 ± 20.18	89.98 ± 10.55	67.48 ± 13.25
BME	40.96 ± 0.33	182.64 ± 78.64	462.14 ± 80.76	24.75 ± 3.34	64.21 ± 8.69	90.85 ± 30.39	378.58 ± 91.95	190.43 ± 66.15
Beef	29.63 ± 0.12	1212.85 ± 418.01	44.79 ± 6.51	25.39 ± 7.57	180.20 ± 140.03	1517.10 ± 851.21	962.64 ± 83.37	37.08 ± 16.03
BeetleFly	4.10 ± 0.00	59.23 ± 0.42	7.39 ± 0.91	6.38 ± 0.80	53.02 ± 6.79	53.98 ± 23.93	54.46 ± 1.85	31.48 ± 7.71
BirdChicken	5.94 ± 0.08	118.07 ± 2.39	17.77 ± 2.11	8.41 ± 2.01	117.47 ± 0.56	107.78 ± 52.19	122.15 ± 6.69	28.36 ± 7.23
CBF	102.11 ± 1.38	329.46 ± 43.82	176.18 ± 9.18	80.65 ± 13.61	332.66 ± 55.38	137.79 ± 33.53	319.67 ± 51.40	308.15 ± 31.66
Car	32.55 ± 0.87	182.98 ± 49.58	36.39 ± 3.58	27.06 ± 7.60	49.70 ± 6.58	29.88 ± 6.65	72.72 ± 12.38	35.82 ± 4.79
Chinatown	102.23 ± 2.69	310.98 ± 110.82	160.53 ± 24.57	34.46 ± 9.65	218.00 ± 110.81	61.59 ± 4.12	337.55 ± 32.16	278.67 ± 48.45
ChlorineConcentration	1270.52 ± 25.26	1934.83 ± 403.04	1589.63 ± 124.18	402.83 ± 21.48	1767.53 ± 220.68	2712.97 ± 1991.95	1810.60 ± 59.14	1775.03 ± 142.86
CinCECGTorso	129.18 ± 0.89	1217.46 ± 464.73	155.92 ± 5.05	126.41 ± 11.05	261.26 ± 38.34	21680.61 ± 26412.26	1753.56 ± 2365.44	824.69 ± 575.77
Coffee	10.21 ± 0.33	74.91 ± 0.70	22.52 ± 1.71	17.65 ± 4.63	75.00 ± 0.09	37.29 ± 6.97	67.44 ± 9.96	39.33 ± 19.52
Computers	21.11 ± 0.18	392.82 ± 7.85	107.84 ± 2.35	47.75 ± 7.70	68.54 ± 13.71	217.59 ± 95.72	316.16 ± 82.40	146.70 ± 24.72
CricketX	47.44 ± 0.58	2657.57 ± 2100.09	47.95 ± 6.45	43.81 ± 2.30	305.70 ± 96.82	788.38 ± 370.96	229.03 ± 80.27	118.27 ± 18.45
CricketY	43.72 ± 1.18	1083.23 ± 1391.25	46.42 ± 1.95	45.78 ± 2.87	114.23 ± 22.79	782.68 ± 338.23	364.89 ± 116.55	109.86 ± 6.41
CricketZ	44.91 ± 0.65	1307.69 ± 1355.67	50.10 ± 1.55	46.59 ± 2.07	126.13 ± 38.93	1287.80 ± 774.12	326.09 ± 138.09	97.10 ± 17.33
Crop	3576.22 ± 253.76	4740.96 ± 602.79	5812.62 ± 767.73	1311.51 ± 137.92	4395.74 ± 712.15	5988.21 ± 3852.82	11202.62 ± 1237.83	16281.47 ± 2552.36
DiatomSizeReduction	207.46 ± 1.21	197.88 ± 47.44	2956.71 ± 655.70	229.03 ± 72.48	232.12 ± 69.76	271.11 ± 169.77	240.38 ± 48.80	256.52 ± 79.62
DistalPhalanxOutlineAgeGroup	256.38 ± 7.73	918.44 ± 630.80	196.52 ± 39.53	64.21 ± 21.28	840.56 ± 264.88	163.48 ± 37.31	635.83 ± 157.88	833.18 ± 144.97
DistalPhalanxOutlineCorrect	419.06 ± 28.46	689.94 ± 197.02	266.65 ± 40.83	76.79 ± 8.26	659.77 ± 67.60	180.54 ± 41.98	960.87 ± 323.30	952.64 ± 166.31
DistalPhalanxTW	132.89 ± 5.96	992.60 ± 316.76	182.51 ± 17.25	50.74 ± 8.47	523.34 ± 114.55	120.35 ± 23.66	763.11 ± 144.47	685.53 ± 149.89
ECG200	59.14 ± 1.03	117.69 ± 29.81	27.16 ± 1.18	35.21 ± 13.81	93.78 ± 14.19	76.63 ± 30.90	123.51 ± 41.77	101.97 ± 32.55
ECG5000	940.62 ± 19.61	1923.72 ± 183.76	920.97 ± 153.75	357.19 ± 25.11	1943.39 ± 450.53	1267.25 ± 489.49	1449.68 ± 249.80	1401.31 ± 114.55
ECGFiveDays	94.23 ± 0.70	245.13 ± 36.56	125.02 ± 6.40	80.43 ± 8.85	163.44 ± 19.70	118.63 ± 26.15	191.62 ± 37.55	188.01 ± 11.23
EOGHorizontalSignal	40.90 ± 2.24	2197.58 ± 0.17	56.59 ± 2.15	49.80 ± 1.57	191.04 ± 9.65	2197.42 ± 0.04	2344.05 ± 572.85	249.34 ± 181.02
EOGVerticalSignal	59.79 ± 3.37	2101.07 ± 23.84	109.80 ± 26.45	75.77 ± 14.68	1850.36 ± 438.64	2067.36 ± 341.45	14737.07 ± 26277.54	662.20 ± 589.28
Earthquakes	1.56 ± 0.01	336.26 ± 33.68	148.73 ± 37.29	26.04 ± 4.32	570.12 ± 16.71	398.72 ± 642.18	630.95 ± 140.28	208.79 ± 36.27
ElectricDevices	205.80 ± 4.77	2434.59 ± 534.75	2437.92 ± 422.36	574.87 ± 52.19	1437.68 ± 97.80	3682.77 ± 1196.05	4373.51 ± 953.11	4131.23 ± 658.92
EthanolLevel	90.13 ± 0.29	1313.48 ± 1292.16	80.49 ± 4.54	89.83 ± 9.91	687.49 ± 304.95	1432.24 ± 375.18	578.09 ± 210.31	610.79 ± 215.12
FaceAll	29.92 ± 3.51	643.23 ± 638.79	130.99 ± 19.08	86.23 ± 4.61	724.12 ± 543.53	825.05 ± 889.09	816.53 ± 159.43	302.28 ± 68.70
FaceFour	10.28 ± 0.22	62.81 ± 13.59	47.76 ± 7.45	18.19 ± 2.49	43.12 ± 8.23	38.94 ± 15.92	83.15 ± 7.54	34.84 ± 7.58
FacesUCR	43.89 ± 4.01	366.55 ± 132.60	117.77 ± 12.02	90.27 ± 10.19	209.52 ± 33.71	451.72 ± 139.61	355.97 ± 34.24	282.78 ± 53.49
FiftyWords	16.04 ± 0.52	306.64 ± 206.88	16.53 ± 0.45	29.60 ± 2.05	70.29 ± 5.45	278.33 ± 80.79	129.90 ± 33.57	56.82 ± 11.26

Continued on next page

Table C3 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
Fish	37.50 ± 0.32	166.38 ± 41.21	52.29 ± 3.40	35.28 ± 8.18	66.84 ± 7.44	150.05 ± 97.95	133.12 ± 21.13	66.34 ± 10.68
FordA	99.59 ± 4.35	1385.37 ± 344.74	759.46 ± 16.38	322.96 ± 31.81	716.64 ± 677.81	3378.38 ± 480.73	3397.21 ± 380.83	2481.79 ± 355.34
FordB	43.60 ± 6.81	1403.24 ± 452.30	429.30 ± 18.24	256.06 ± 17.97	1492.93 ± 879.70	2842.84 ± 1414.37	3057.92 ± 106.78	1558.49 ± 144.65
FreezerRegularTrain	1357.32 ± 8.27	2365.61 ± 674.45	1340.27 ± 164.83	306.22 ± 33.85	3573.46 ± 155.38	515.70 ± 182.94	1845.31 ± 279.12	1893.76 ± 139.33
FreezerSmallTrain	1535.63 ± 12.54	2129.89 ± 588.97	1150.65 ± 187.99	278.71 ± 12.96	2739.45 ± 753.94	626.99 ± 135.89	1550.30 ± 137.16	1537.79 ± 212.49
Fungi	22.00 ± 0.57	233.52 ± 132.04	15.02 ± 0.47	25.35 ± 5.57	55.20 ± 4.35	153.28 ± 66.93	162.26 ± 43.95	32.46 ± 5.21
GunPoint	73.16 ± 25.07	198.39 ± 81.66	46.47 ± 3.48	27.86 ± 8.57	208.26 ± 29.28	56.54 ± 11.53	169.77 ± 47.43	121.35 ± 52.16
GunPointAgeSpan	816.01 ± 0.00	2432.25 ± 2206.51	679.39 ± 76.15	62.64 ± 5.10	614.35 ± 177.81	374.09 ± 236.73	665.01 ± 83.99	719.45 ± 93.70
GunPointMaleVersusFemale	875.76 ± 0.00	2996.61 ± 2229.31	609.62 ± 58.43	88.48 ± 20.16	613.22 ± 281.03	702.90 ± 553.97	802.20 ± 179.00	700.82 ± 121.09
GunPointOldVersusYoung	1017.57 ± 0.00	3360.71 ± 1259.68	924.98 ± 252.28	171.13 ± 120.25	705.58 ± 329.87	1213.48 ± 587.20	1000.60 ± 174.62	849.74 ± 131.24
Ham	8.40 ± 0.70	63.67 ± 19.85	43.00 ± 2.20	21.29 ± 1.57	68.23 ± 25.33	51.90 ± 23.65	75.13 ± 23.70	39.28 ± 3.52
Haptics	60.02 ± 0.89	297.42 ± 136.02	34.53 ± 6.45	41.92 ± 6.60	127.08 ± 30.98	4025.27 ± 7581.81	336.20 ± 272.49	64.85 ± 22.88
Herring	23.97 ± 1.68	95.94 ± 19.16	30.93 ± 1.25	17.56 ± 4.03	41.88 ± 8.53	62.74 ± 23.47	103.88 ± 8.85	26.50 ± 5.92
HouseTwenty	10.19 ± 0.00	269.31 ± 46.04	41.21 ± 11.04	30.44 ± 14.58	269.24 ± 7.98	201.52 ± 79.36	221.67 ± 46.25	94.52 ± 52.97
InlineSkate	77.53 ± 0.52	1116.93 ± 64.69	107.62 ± 26.13	64.43 ± 3.74	210.84 ± 218.23	1197.76 ± 286.92	603.82 ± 267.57	132.41 ± 29.40
InsectEPGRegularTrain	14393.62 ± 0.00	13544.35 ± 331.53	8881.58 ± 984.90	4539.69 ± 1636.80	14744.91 ± 49.29	47071.32 ± 68200.74	17562.31 ± 1576.79	11455.13 ± 1428.16
InsectEPGSmallTrain	11887.39 ± 0.00	10347.89 ± 543.92	5855.65 ± 913.47	1386.40 ± 1541.13	11188.37 ± 176.09	835415.32 ± 1348449.69	11934.29 ± 1621.46	8552.22 ± 892.29
InsectWingbeatSound	89.89 ± 10.31	1255.08 ± 566.93	146.93 ± 3.75	139.33 ± 9.75	542.88 ± 79.21	972.96 ± 426.95	431.85 ± 32.06	355.56 ± 29.21
ItalyPowerDemand	434.72 ± 16.57	1093.36 ± 229.66	245.92 ± 39.15	93.96 ± 9.45	1129.04 ± 285.70	193.43 ± 13.15	458.23 ± 88.68	563.53 ± 131.01
LargeKitchenAppliances	8.52 ± 0.39	1459.60 ± 489.35	629.69 ± 69.42	71.97 ± 25.05	452.74 ± 229.48	691.60 ± 276.74	570.70 ± 237.38	177.84 ± 64.72
Lightning2	8.50 ± 0.17	82.02 ± 23.60	133.91 ± 4.54	18.04 ± 2.35	64.46 ± 7.62	39.61 ± 17.91	61.30 ± 15.90	75.42 ± 17.05
Lightning7	10.85 ± 0.37	248.41 ± 107.10	46.59 ± 4.56	19.25 ± 2.81	74.47 ± 6.02	87.94 ± 20.25	171.18 ± 51.35	56.09 ± 10.27
Mallat	196.53 ± 5.64	2450.76 ± 526.34	114.11 ± 2.21	139.17 ± 16.80	681.62 ± 1060.76	4028.21 ± 1788.94	1509.89 ± 645.83	472.06 ± 341.06
Meat	31.42 ± 1.92	124.33 ± 34.75	64.30 ± 11.59	23.22 ± 3.15	80.98 ± 18.75	109.48 ± 46.80	54.84 ± 15.11	38.20 ± 7.31
MedicalImages	70.41 ± 6.66	207.66 ± 22.00	155.84 ± 8.42	87.08 ± 6.91	253.67 ± 16.57	157.22 ± 37.77	290.63 ± 42.96	284.52 ± 42.79
MiddlePhalanxOutlineAgeGroup	235.09 ± 11.89	553.85 ± 102.31	228.01 ± 24.82	53.25 ± 5.92	494.29 ± 127.79	135.72 ± 40.58	535.31 ± 83.13	668.62 ± 31.17
MiddlePhalanxOutlineCorrect	260.29 ± 7.19	283.82 ± 45.15	271.17 ± 29.74	78.74 ± 8.92	507.85 ± 158.97	246.79 ± 47.06	617.12 ± 155.28	674.46 ± 164.56
MiddlePhalanxTW	124.32 ± 3.74	416.95 ± 109.66	177.41 ± 16.94	52.04 ± 16.45	691.33 ± 264.08	132.73 ± 38.03	356.09 ± 40.26	348.81 ± 73.20
MixedShapesRegularTrain	250.04 ± 8.65	1078.63 ± 1146.46	250.42 ± 2.72	256.58 ± 22.31	369.83 ± 35.36	4634.49 ± 1093.49	1777.01 ± 1314.89	675.91 ± 354.75
MixedShapesSmallTrain	222.81 ± 6.89	2194.41 ± 1780.04	223.76 ± 10.37	214.10 ± 9.12	358.75 ± 52.90	5039.43 ± 2068.30	2378.67 ± 1275.04	577.14 ± 257.75
MoteStrain	66.01 ± 12.66	529.38 ± 89.48	175.93 ± 18.86	108.62 ± 12.58	762.44 ± 268.14	128.74 ± 14.15	359.45 ± 63.53	432.87 ± 103.16
NonInvasiveFetalECGThorax1	116.37 ± 4.22	1692.81 ± 2013.36	113.83 ± 9.02	147.05 ± 11.41	219.54 ± 65.70	6949.35 ± 5327.56	304.73 ± 34.76	202.75 ± 24.11

Continued on next page

Table C3 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
NonInvasiveFetalECGThorax2	108.45 ± 3.85	201.38 ± 40.55	135.59 ± 18.38	165.80 ± 13.70	196.98 ± 20.31	1640.75 ± 646.38	247.46 ± 15.25	220.10 ± 38.34
OSULeaf	22.79 ± 0.82	536.56 ± 240.02	28.41 ± 1.88	32.79 ± 2.15	159.74 ± 19.20	80.13 ± 26.68	1109.81 ± 986.35	111.63 ± 12.73
OliveOil	20.80 ± 1.08	234.24 ± 3.97	15.93 ± 0.95	22.68 ± 6.26	129.15 ± 78.20	181.71 ± 63.69	232.01 ± 2.13	289.27 ± 519.77
PhalangesOutlinesCorrect	713.74 ± 55.79	1422.08 ± 381.39	651.02 ± 122.60	206.59 ± 24.48	1713.95 ± 953.47	584.09 ± 269.92	1082.19 ± 344.82	1517.80 ± 590.63
Phoneme	2.80 ± 0.11	161.39 ± 142.82	31.82 ± 0.87	30.11 ± 2.42	63.46 ± 8.21	423.81 ± 263.05	259.25 ± 88.22	229.99 ± 192.89
PigAirwayPressure	33.24 ± 1.09	17621.79 ± 2986.64	42.49 ± 2.01	43.71 ± 5.39	22346.02 ± 1182.10	6448.94 ± 4965.81	10921.98 ± 7018.33	9966.10 ± 6292.60
PigArtPressure	4.56 ± 0.21	4204.80 ± 1017.83	36.22 ± 0.72	15.47 ± 0.99	538.06 ± 369.68	1988.19 ± 2054.40	491.14 ± 258.15	430.79 ± 339.58
PigCVP	5.90 ± 2.37	2529.64 ± 995.25	496.42 ± 102.12	18.64 ± 1.03	437.03 ± 726.10	1340.27 ± 717.44	2854.01 ± 3977.08	976.59 ± 1821.02
Plane	84.89 ± 7.11	278.77 ± 45.91	47.89 ± 5.64	46.83 ± 11.79	324.25 ± 93.59	60.38 ± 9.12	286.56 ± 56.02	170.18 ± 21.83
PowerCons	52.32 ± 0.95	1609.13 ± 1143.69	113.41 ± 4.12	37.81 ± 9.98	222.43 ± 28.75	79.14 ± 16.10	223.98 ± 40.79	194.12 ± 15.23
ProximalPhalanxOutlineAgeGroup	264.42 ± 15.26	931.82 ± 394.43	274.78 ± 31.04	63.18 ± 9.31	1073.35 ± 172.50	123.77 ± 25.29	923.75 ± 356.41	1113.26 ± 210.42
ProximalPhalanxOutlineCorrect	625.64 ± 9.50	950.90 ± 639.27	363.89 ± 62.96	90.34 ± 9.28	1268.07 ± 376.44	232.61 ± 92.42	608.29 ± 144.12	809.12 ± 304.92
ProximalPhalanxTW	170.91 ± 10.19	615.71 ± 126.54	162.33 ± 21.38	59.82 ± 7.78	979.74 ± 414.93	120.64 ± 38.41	843.83 ± 280.23	704.32 ± 87.27
RefrigerationDevices	2.31 ± 0.07	958.00 ± 272.99	85.71 ± 7.46	36.99 ± 4.89	348.20 ± 59.14	1070.67 ± 1259.64	138.22 ± 56.28	309.15 ± 42.41
Rock	23.00 ± 0.65	158.36 ± 19.61	53.25 ± 4.93	19.15 ± 3.62	121.71 ± 17.94	209.37 ± 72.24	206.52 ± 109.95	56.35 ± 30.87
ScreenType	29.73 ± 0.44	847.34 ± 97.04	110.23 ± 11.10	61.85 ± 9.72	113.96 ± 8.03	452.97 ± 153.93	524.32 ± 183.58	158.89 ± 45.06
SemgHandGenderCh2	252.36 ± 0.00	1303.29 ± 658.79	508.18 ± 62.74	177.96 ± 59.56	419.86 ± 44.78	2572.77 ± 2670.11	807.56 ± 210.77	469.77 ± 75.16
SemgHandMovementCh2	83.03 ± 0.66	1905.65 ± 1023.34	453.12 ± 23.52	125.75 ± 13.99	258.39 ± 46.42	1728.12 ± 1082.02	816.06 ± 426.83	227.77 ± 17.14
SemgHandSubjectCh2	101.15 ± 0.20	1813.92 ± 487.70	481.21 ± 35.12	127.36 ± 8.06	332.97 ± 48.74	1496.57 ± 948.48	752.76 ± 270.49	335.38 ± 103.66
ShapeletSim	1.14 ± 0.01	220.04 ± 5.65	33.79 ± 5.08	10.31 ± 1.55	223.20 ± 2.00	103.31 ± 72.83	249.00 ± 19.61	143.72 ± 13.56
ShapesAll	34.87 ± 0.80	300.98 ± 246.26	24.73 ± 1.17	50.08 ± 4.18	85.42 ± 10.46	1422.50 ± 811.99	475.02 ± 59.18	83.88 ± 17.11
SmallKitchenAppliances	16.81 ± 2.02	242.37 ± 32.48	878.05 ± 35.87	53.38 ± 7.21	204.48 ± 39.59	674.97 ± 131.90	534.87 ± 327.67	244.90 ± 35.22
SmoothSubspace	16.68 ± 0.74	502.93 ± 150.33	59.82 ± 5.61	27.92 ± 2.34	240.28 ± 87.05	92.47 ± 24.95	415.44 ± 162.27	201.77 ± 88.83
SonyAIBORobotSurface1	30.82 ± 2.25	218.12 ± 51.41	99.96 ± 14.13	49.77 ± 6.94	186.89 ± 52.17	88.21 ± 18.84	235.74 ± 100.87	148.52 ± 13.52
SonyAIBORobotSurface2	140.14 ± 18.82	1240.63 ± 533.93	95.13 ± 11.39	87.46 ± 13.25	1117.32 ± 231.08	169.89 ± 13.37	661.62 ± 286.52	334.52 ± 93.95
StarLightCurves	5613.60 ± 24.23	8393.80 ± 3098.03	2070.60 ± 143.99	752.96 ± 74.51	4591.01 ± 2604.00	14816.50 ± 18168.33	15279.79 ± 9886.39	12506.95 ± 5139.68
Strawberry	293.56 ± 2.55	349.45 ± 80.62	425.25 ± 21.27	115.66 ± 14.19	315.24 ± 47.69	111.76 ± 23.18	373.53 ± 15.27	408.76 ± 41.56
SwedishLeaf	44.22 ± 0.86	158.60 ± 24.41	97.32 ± 9.17	60.52 ± 3.24	121.30 ± 13.24	78.88 ± 7.07	330.04 ± 36.26	251.39 ± 42.37
Symbols	295.00 ± 13.13	7235.79 ± 4928.51	488.74 ± 53.25	157.69 ± 15.66	995.07 ± 268.88	15893.09 ± 13026.99	4737.76 ± 1184.24	1427.63 ± 96.92
SyntheticControl	52.39 ± 0.58	1019.55 ± 594.45	225.02 ± 8.83	42.53 ± 3.99	1208.12 ± 801.82	83.95 ± 16.50	1530.31 ± 663.15	1023.76 ± 152.23
ToeSegmentation1	15.61 ± 3.16	225.27 ± 9.21	51.80 ± 8.37	25.63 ± 3.53	210.85 ± 14.27	127.53 ± 61.28	216.10 ± 15.03	65.86 ± 11.77
ToeSegmentation2	14.75 ± 0.93	157.07 ± 0.84	27.98 ± 5.24	17.40 ± 1.74	156.67 ± 0.91	75.04 ± 21.81	136.59 ± 30.47	53.33 ± 9.42

Continued on next page

Table C3 – *Continued from previous page*

Dataset	Raw Data	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
Trace	133.63 ± 1.50	915.26 ± 1118.23	70.32 ± 8.19	36.22 ± 2.93	557.86 ± 283.68	269.86 ± 190.79	359.08 ± 69.95	261.07 ± 58.66
TwoLeadECG	304.88 ± 41.98	424.74 ± 123.10	325.83 ± 25.15	124.70 ± 9.47	474.95 ± 134.63	184.47 ± 32.04	607.00 ± 81.06	644.90 ± 103.06
TwoPatterns	134.53 ± 10.18	3261.84 ± 540.08	597.56 ± 45.62	281.11 ± 25.79	4677.33 ± 5578.97	1360.44 ± 775.11	2182.31 ± 492.06	1337.50 ± 140.69
UMD	34.60 ± 0.77	58.87 ± 5.35	208.07 ± 16.08	33.99 ± 5.70	46.87 ± 6.56	119.39 ± 104.41	314.24 ± 149.05	148.52 ± 13.37
UWaveGestureLibraryAll	214.36 ± 1.65	3422.33 ± 2628.67	128.07 ± 4.30	309.14 ± 24.62	658.19 ± 358.97	12692.25 ± 5734.10	2386.08 ± 2330.83	628.99 ± 457.70
UWaveGestureLibraryX	401.83 ± 18.52	876.26 ± 183.98	242.19 ± 16.65	342.59 ± 13.45	446.76 ± 60.30	2476.73 ± 1335.96	593.92 ± 73.96	734.17 ± 327.65
UWaveGestureLibraryY	618.86 ± 15.13	720.85 ± 168.63	277.75 ± 5.40	358.56 ± 41.29	542.03 ± 89.02	1884.45 ± 582.29	624.05 ± 16.56	679.71 ± 75.15
UWaveGestureLibraryZ	457.57 ± 11.47	795.90 ± 64.95	270.29 ± 12.61	359.78 ± 28.90	526.44 ± 59.18	2590.14 ± 428.03	651.74 ± 82.19	569.68 ± 54.93
Wafer	6429.07 ± 4.80	5456.71 ± 1244.88	2004.10 ± 192.31	669.64 ± 54.47	8953.39 ± 1702.72	2214.34 ± 759.89	2090.32 ± 181.30	1902.95 ± 209.46
Wine	18.69 ± 2.93	55.94 ± 13.78	41.02 ± 2.10	18.71 ± 3.79	33.06 ± 5.19	133.25 ± 192.82	111.66 ± 34.72	48.29 ± 11.50
WordSynonyms	20.10 ± 0.93	269.52 ± 59.45	25.22 ± 2.80	40.03 ± 3.83	95.94 ± 20.68	162.90 ± 62.43	101.93 ± 15.25	65.83 ± 5.51
Worms	10.07 ± 0.12	394.48 ± 6.15	22.74 ± 3.37	18.98 ± 1.48	103.10 ± 68.16	1252.17 ± 227.58	149.01 ± 118.86	57.47 ± 3.39
WormsTwoClass	17.40 ± 0.14	268.58 ± 1.22	32.79 ± 4.33	23.65 ± 1.19	50.21 ± 6.22	404.59 ± 353.53	191.95 ± 86.71	47.24 ± 4.81
Yoga	859.80 ± 49.86	1429.04 ± 964.82	494.60 ± 50.15	331.77 ± 48.57	973.13 ± 144.93	2286.78 ± 562.95	731.20 ± 83.56	650.72 ± 101.61

Table C4: Detailed results of the evaluations of of the experiments and the baselines using Accuracy on the downstream task of time series classification. The number denotes (mean ± standard deviation) of the Accuracy metric from 5 runs of each experiment. The rows represent the datasets and the columns represent the model, either a baseline or a model trained within the TSRC framework. The naming convention used in this table for the TSRC framework is as follows: {student model name} w/ TSRC ({teacher model name}).

Dataset	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
ACSF1	0.62 ± 0.04	0.60 ± 0.01	0.64 ± 0.05	0.61 ± 0.02	0.59 ± 0.04	0.63 ± 0.03	0.62 ± 0.05
Adiac	0.54 ± 0.06	0.68 ± 0.02	0.69 ± 0.02	0.60 ± 0.04	0.51 ± 0.06	0.57 ± 0.02	0.66 ± 0.02
ArrowHead	0.54 ± 0.04	0.87 ± 0.02	0.86 ± 0.02	0.59 ± 0.06	0.66 ± 0.08	0.65 ± 0.06	0.72 ± 0.06
BME	0.75 ± 0.04	0.94 ± 0.01	0.97 ± 0.02	0.88 ± 0.08	0.79 ± 0.05	0.83 ± 0.05	0.96 ± 0.03
Beef	0.28 ± 0.05	0.39 ± 0.07	0.54 ± 0.06	0.45 ± 0.03	0.39 ± 0.08	0.31 ± 0.08	0.52 ± 0.03
BeetleFly	0.70 ± 0.00	0.82 ± 0.12	0.70 ± 0.04	0.73 ± 0.06	0.67 ± 0.06	0.70 ± 0.00	0.72 ± 0.05
BirdChicken	0.34 ± 0.02	0.73 ± 0.04	0.66 ± 0.02	0.33 ± 0.02	0.42 ± 0.05	0.35 ± 0.03	0.63 ± 0.08
CBF	0.97 ± 0.02	1.00 ± 0.00	0.98 ± 0.01	0.98 ± 0.02	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.00
Car	0.29 ± 0.06	0.68 ± 0.03	0.68 ± 0.04	0.43 ± 0.05	0.44 ± 0.07	0.39 ± 0.04	0.56 ± 0.04
Chinatown	0.97 ± 0.01	0.96 ± 0.01	0.96 ± 0.01	0.95 ± 0.01	0.95 ± 0.02	0.96 ± 0.02	0.97 ± 0.00
ChlorineConcentration	0.99 ± 0.01	0.99 ± 0.00	0.98 ± 0.01	1.00 ± 0.00	0.78 ± 0.13	0.93 ± 0.01	0.97 ± 0.01
CinCECGTorso	0.55 ± 0.04	1.00 ± 0.00	0.94 ± 0.02	0.91 ± 0.04	0.64 ± 0.09	0.75 ± 0.20	0.80 ± 0.11
Coffee	0.50 ± 0.00	0.72 ± 0.19	0.85 ± 0.04	0.50 ± 0.00	0.64 ± 0.17	0.56 ± 0.13	0.61 ± 0.16
Computers	0.60 ± 0.00	0.60 ± 0.03	0.56 ± 0.04	0.64 ± 0.02	0.58 ± 0.02	0.62 ± 0.01	0.60 ± 0.02
CricketX	0.17 ± 0.02	0.68 ± 0.01	0.54 ± 0.02	0.42 ± 0.03	0.26 ± 0.06	0.39 ± 0.07	0.57 ± 0.03
CricketY	0.18 ± 0.01	0.68 ± 0.02	0.49 ± 0.02	0.34 ± 0.02	0.28 ± 0.04	0.33 ± 0.06	0.61 ± 0.03
CricketZ	0.17 ± 0.02	0.73 ± 0.03	0.57 ± 0.01	0.24 ± 0.04	0.27 ± 0.05	0.34 ± 0.07	0.45 ± 0.08
Crop	0.71 ± 0.01	0.71 ± 0.01	0.69 ± 0.01	0.72 ± 0.01	0.67 ± 0.03	0.70 ± 0.01	0.71 ± 0.01

Continued on next page

Table C4 – *Continued from previous page*

Dataset	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
DiatomSizeReduction	0.88 ± 0.02	0.96 ± 0.00	0.97 ± 0.01	0.98 ± 0.02	0.94 ± 0.01	0.97 ± 0.01	0.96 ± 0.00
DistalPhalanxOutlineAgeGroup	0.79 ± 0.02	0.77 ± 0.00	0.76 ± 0.03	0.78 ± 0.02	0.79 ± 0.02	0.79 ± 0.02	0.79 ± 0.02
DistalPhalanxOutlineCorrect	0.78 ± 0.03	0.82 ± 0.01	0.79 ± 0.02	0.78 ± 0.01	0.75 ± 0.02	0.77 ± 0.02	0.77 ± 0.02
DistalPhalanxTW	0.75 ± 0.01	0.72 ± 0.01	0.71 ± 0.02	0.75 ± 0.01	0.71 ± 0.02	0.75 ± 0.01	0.75 ± 0.01
ECG200	0.79 ± 0.04	0.83 ± 0.01	0.87 ± 0.03	0.83 ± 0.04	0.82 ± 0.03	0.80 ± 0.02	0.85 ± 0.02
ECG5000	0.95 ± 0.00	0.95 ± 0.00	0.95 ± 0.00	0.95 ± 0.00	0.94 ± 0.00	0.95 ± 0.00	0.95 ± 0.00
ECGFiveDays	0.97 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.01	0.94 ± 0.02	0.96 ± 0.02	0.98 ± 0.01
EOGHorizontalSignal	0.15 ± 0.00	0.70 ± 0.01	0.58 ± 0.04	0.50 ± 0.04	0.16 ± 0.00	0.18 ± 0.01	0.50 ± 0.12
EOGVerticalSignal	0.18 ± 0.00	0.62 ± 0.02	0.48 ± 0.04	0.25 ± 0.06	0.22 ± 0.03	0.20 ± 0.03	0.34 ± 0.08
Earthquakes	0.82 ± 0.00	0.79 ± 0.01	0.80 ± 0.00	0.81 ± 0.01	0.80 ± 0.01	0.80 ± 0.01	0.80 ± 0.01
ElectricDevices	0.70 ± 0.01	0.79 ± 0.01	0.60 ± 0.01	0.72 ± 0.01	0.67 ± 0.04	0.78 ± 0.01	0.76 ± 0.01
EthanolLevel	0.36 ± 0.05	0.45 ± 0.01	0.40 ± 0.06	0.40 ± 0.03	0.37 ± 0.01	0.39 ± 0.03	0.41 ± 0.03
FaceAll	0.65 ± 0.10	0.92 ± 0.01	0.80 ± 0.01	0.74 ± 0.08	0.53 ± 0.13	0.71 ± 0.04	0.92 ± 0.02
FaceFour	0.57 ± 0.07	0.91 ± 0.02	0.88 ± 0.02	0.67 ± 0.05	0.60 ± 0.09	0.53 ± 0.01	0.81 ± 0.06
FacesUCR	0.80 ± 0.04	0.92 ± 0.01	0.79 ± 0.02	0.87 ± 0.02	0.63 ± 0.02	0.84 ± 0.00	0.93 ± 0.01
FiftyWords	0.41 ± 0.02	0.74 ± 0.01	0.61 ± 0.03	0.59 ± 0.01	0.37 ± 0.06	0.45 ± 0.03	0.61 ± 0.06
Fish	0.39 ± 0.03	0.81 ± 0.02	0.79 ± 0.03	0.51 ± 0.04	0.41 ± 0.03	0.47 ± 0.03	0.63 ± 0.06
FordA	0.64 ± 0.01	0.91 ± 0.00	0.65 ± 0.01	0.72 ± 0.05	0.59 ± 0.02	0.64 ± 0.02	0.86 ± 0.03
FordB	0.66 ± 0.02	0.88 ± 0.00	0.57 ± 0.02	0.67 ± 0.03	0.57 ± 0.04	0.63 ± 0.03	0.81 ± 0.04
FreezerRegularTrain	0.98 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.78 ± 0.03	0.99 ± 0.00	0.99 ± 0.00
FreezerSmallTrain	0.97 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.87 ± 0.05	0.99 ± 0.00	0.99 ± 0.00
Fungi	0.46 ± 0.04	0.93 ± 0.00	0.93 ± 0.05	0.70 ± 0.09	0.55 ± 0.07	0.41 ± 0.04	0.65 ± 0.14
GunPoint	0.57 ± 0.12	0.54 ± 0.11	0.92 ± 0.02	0.52 ± 0.31	0.74 ± 0.09	0.67 ± 0.20	0.57 ± 0.15
GunPointAgeSpan	0.89 ± 0.05	0.97 ± 0.00	0.95 ± 0.02	0.92 ± 0.02	0.91 ± 0.03	0.90 ± 0.01	0.94 ± 0.02
GunPointMaleVersusFemale	0.85 ± 0.04	1.00 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	0.92 ± 0.05	0.97 ± 0.01	0.98 ± 0.01
GunPointOldVersusYoung	0.96 ± 0.04	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Ham	0.52 ± 0.02	0.64 ± 0.02	0.74 ± 0.02	0.52 ± 0.03	0.54 ± 0.03	0.53 ± 0.05	0.64 ± 0.02
Haptics	0.35 ± 0.03	0.44 ± 0.02	0.38 ± 0.01	0.40 ± 0.05	0.35 ± 0.03	0.39 ± 0.05	0.43 ± 0.04
Herring	0.61 ± 0.00	0.61 ± 0.00	0.60 ± 0.07	0.61 ± 0.00	0.61 ± 0.00	0.61 ± 0.00	0.61 ± 0.00
HouseTwenty	0.68 ± 0.03	0.91 ± 0.02	0.73 ± 0.09	0.65 ± 0.03	0.65 ± 0.04	0.67 ± 0.03	0.72 ± 0.09
InlineSkate	0.38 ± 0.04	0.53 ± 0.00	0.40 ± 0.03	0.44 ± 0.04	0.43 ± 0.08	0.36 ± 0.09	0.49 ± 0.03
InsectEPGRegularTrain	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
InsectEPGSmallTrain	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
InsectWingbeatSound	0.52 ± 0.10	0.67 ± 0.00	0.65 ± 0.01	0.63 ± 0.01	0.48 ± 0.06	0.60 ± 0.01	0.66 ± 0.01
ItalyPowerDemand	0.95 ± 0.01	0.94 ± 0.01	0.96 ± 0.00	0.96 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.01
LargeKitchenAppliances	0.59 ± 0.01	0.74 ± 0.01	0.51 ± 0.02	0.58 ± 0.02	0.49 ± 0.04	0.52 ± 0.04	0.69 ± 0.02
Lightning2	0.64 ± 0.02	0.62 ± 0.02	0.74 ± 0.02	0.67 ± 0.02	0.64 ± 0.03	0.66 ± 0.03	0.64 ± 0.02
Lightning7	0.42 ± 0.05	0.68 ± 0.03	0.65 ± 0.04	0.61 ± 0.02	0.56 ± 0.04	0.55 ± 0.05	0.65 ± 0.03
Mallat	0.33 ± 0.01	1.00 ± 0.00	0.91 ± 0.03	0.89 ± 0.13	0.63 ± 0.06	0.57 ± 0.35	0.95 ± 0.02
Meat	0.70 ± 0.04	0.96 ± 0.01	0.95 ± 0.02	0.94 ± 0.03	0.78 ± 0.06	0.92 ± 0.04	0.97 ± 0.01
MedicalImages	0.66 ± 0.02	0.75 ± 0.01	0.69 ± 0.04	0.70 ± 0.03	0.62 ± 0.03	0.68 ± 0.01	0.73 ± 0.01
MiddlePhalanxOutlineAgeGroup	0.72 ± 0.02	0.71 ± 0.03	0.73 ± 0.01	0.74 ± 0.02	0.73 ± 0.00	0.75 ± 0.01	0.72 ± 0.03
MiddlePhalanxOutlineCorrect	0.76 ± 0.02	0.75 ± 0.01	0.77 ± 0.01	0.77 ± 0.01	0.71 ± 0.02	0.73 ± 0.02	0.76 ± 0.02
MiddlePhalanxTW	0.62 ± 0.01	0.59 ± 0.00	0.58 ± 0.05	0.63 ± 0.01	0.60 ± 0.01	0.61 ± 0.01	0.62 ± 0.02
MixedShapesRegularTrain	0.80 ± 0.10	0.96 ± 0.00	0.89 ± 0.01	0.92 ± 0.00	0.66 ± 0.02	0.84 ± 0.09	0.91 ± 0.02
MixedShapesSmallTrain	0.76 ± 0.07	0.96 ± 0.00	0.89 ± 0.02	0.93 ± 0.01	0.68 ± 0.21	0.80 ± 0.08	0.92 ± 0.02
MoteStrain	0.91 ± 0.01	0.91 ± 0.02	0.92 ± 0.01	0.91 ± 0.02	0.91 ± 0.01	0.91 ± 0.01	0.92 ± 0.01
NonInvasiveFetalECGThorax1	0.57 ± 0.07	0.88 ± 0.00	0.76 ± 0.03	0.81 ± 0.04	0.58 ± 0.03	0.79 ± 0.02	0.85 ± 0.03
NonInvasiveFetalECGThorax2	0.82 ± 0.03	0.90 ± 0.00	0.78 ± 0.01	0.88 ± 0.01	0.71 ± 0.06	0.84 ± 0.02	0.89 ± 0.02
OSULeaf	0.46 ± 0.03	0.73 ± 0.02	0.53 ± 0.02	0.55 ± 0.02	0.49 ± 0.01	0.47 ± 0.04	0.61 ± 0.02
OliveOil	0.41 ± 0.07	0.67 ± 0.03	0.86 ± 0.04	0.61 ± 0.12	0.51 ± 0.06	0.53 ± 0.00	0.76 ± 0.04
PhalangesOutlinesCorrect	0.76 ± 0.01	0.79 ± 0.01	0.78 ± 0.01	0.77 ± 0.01	0.73 ± 0.02	0.76 ± 0.01	0.77 ± 0.01
Phoneme	0.15 ± 0.00	0.34 ± 0.01	0.13 ± 0.01	0.15 ± 0.00	0.14 ± 0.01	0.17 ± 0.03	0.21 ± 0.04
PigAirwayPressure	0.07 ± 0.00	0.40 ± 0.04	0.13 ± 0.03	0.08 ± 0.02	0.07 ± 0.01	0.07 ± 0.00	0.08 ± 0.02
PigArtPressure	0.05 ± 0.02	0.85 ± 0.01	0.26 ± 0.03	0.07 ± 0.03	0.09 ± 0.02	0.10 ± 0.01	0.09 ± 0.03
PigCVP	0.06 ± 0.01	0.32 ± 0.03	0.10 ± 0.02	0.12 ± 0.03	0.06 ± 0.02	0.08 ± 0.06	0.11 ± 0.03
Plane	0.95 ± 0.03	0.99 ± 0.00	0.98 ± 0.01	0.97 ± 0.02	0.95 ± 0.03	0.98 ± 0.02	0.99 ± 0.01
PowerCons	0.75 ± 0.06	0.93 ± 0.01	0.95 ± 0.03	0.94 ± 0.01	0.96 ± 0.02	0.89 ± 0.02	0.94 ± 0.01
ProximalPhalanxOutlineAgeGroup	0.83 ± 0.01	0.78 ± 0.00	0.80 ± 0.01	0.83 ± 0.01	0.80 ± 0.01	0.81 ± 0.02	0.82 ± 0.01
ProximalPhalanxOutlineCorrect	0.77 ± 0.01	0.78 ± 0.03	0.80 ± 0.01	0.78 ± 0.01	0.75 ± 0.02	0.77 ± 0.01	0.79 ± 0.01
ProximalPhalanxTW	0.81 ± 0.01	0.78 ± 0.01	0.81 ± 0.01	0.79 ± 0.01	0.76 ± 0.02	0.79 ± 0.01	0.81 ± 0.01
RefrigerationDevices	0.55 ± 0.01	0.66 ± 0.01	0.37 ± 0.02	0.53 ± 0.01	0.50 ± 0.03	0.49 ± 0.02	0.55 ± 0.02
Rock	0.42 ± 0.06	0.69 ± 0.05	0.67 ± 0.02	0.49 ± 0.04	0.50 ± 0.07	0.48 ± 0.07	0.59 ± 0.06
ScreenType	0.42 ± 0.02	0.44 ± 0.01	0.43 ± 0.02	0.43 ± 0.01	0.41 ± 0.02	0.42 ± 0.02	0.44 ± 0.04

Continued on next page

Table C4 – *Continued from previous page*

Dataset	LSTM-AE	TS2Vec	MCL	TimeNet	LSTM-AE w/ TSRC (MCL)	LSTM-AE w/ TSRC (TS2Vec)	TimeNet w/ TSRC (TS2Vec)
SemgHandGenderCh2	0.75 ± 0.02	0.94 ± 0.01	0.89 ± 0.03	0.90 ± 0.01	0.73 ± 0.03	0.67 ± 0.09	0.88 ± 0.03
SemgHandMovementCh2	0.33 ± 0.02	0.84 ± 0.01	0.71 ± 0.03	0.64 ± 0.03	0.42 ± 0.06	0.44 ± 0.14	0.63 ± 0.10
SemgHandSubjectCh2	0.49 ± 0.02	0.91 ± 0.01	0.80 ± 0.03	0.80 ± 0.03	0.52 ± 0.06	0.61 ± 0.11	0.83 ± 0.03
ShapeletSim	0.52 ± 0.01	1.00 ± 0.00	0.53 ± 0.03	0.52 ± 0.01	0.45 ± 0.09	0.47 ± 0.02	0.45 ± 0.04
ShapesAll	0.51 ± 0.06	0.84 ± 0.01	0.69 ± 0.03	0.72 ± 0.02	0.47 ± 0.03	0.60 ± 0.03	0.76 ± 0.02
SmallKitchenAppliances	0.63 ± 0.01	0.59 ± 0.03	0.52 ± 0.01	0.63 ± 0.02	0.62 ± 0.04	0.61 ± 0.01	0.64 ± 0.02
SmoothSubspace	0.86 ± 0.04	0.70 ± 0.03	0.82 ± 0.02	0.86 ± 0.02	0.83 ± 0.02	0.81 ± 0.03	0.85 ± 0.02
SonyAIBORobotSurface1	0.95 ± 0.02	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.00	0.95 ± 0.02	0.88 ± 0.05	0.96 ± 0.02
SonyAIBORobotSurface2	0.91 ± 0.01	0.89 ± 0.01	0.92 ± 0.02	0.91 ± 0.01	0.87 ± 0.02	0.88 ± 0.02	0.93 ± 0.01
StarLightCurves	0.94 ± 0.00	0.98 ± 0.00	0.94 ± 0.01	0.97 ± 0.00	0.84 ± 0.02	0.92 ± 0.03	0.93 ± 0.04
Strawberry	0.88 ± 0.02	0.95 ± 0.01	0.95 ± 0.00	0.90 ± 0.02	0.90 ± 0.02	0.90 ± 0.01	0.92 ± 0.01
SwedishLeaf	0.73 ± 0.02	0.88 ± 0.01	0.79 ± 0.02	0.82 ± 0.01	0.71 ± 0.03	0.79 ± 0.01	0.87 ± 0.00
Symbols	0.82 ± 0.13	0.98 ± 0.00	0.94 ± 0.01	0.95 ± 0.01	0.84 ± 0.04	0.92 ± 0.02	0.97 ± 0.01
SyntheticControl	0.93 ± 0.02	0.99 ± 0.00	0.90 ± 0.01	0.96 ± 0.03	0.94 ± 0.01	0.92 ± 0.04	0.97 ± 0.02
ToeSegmentation1	0.67 ± 0.02	0.92 ± 0.01	0.65 ± 0.06	0.67 ± 0.05	0.62 ± 0.06	0.59 ± 0.04	0.63 ± 0.05
ToeSegmentation2	0.75 ± 0.00	0.91 ± 0.02	0.81 ± 0.02	0.75 ± 0.00	0.72 ± 0.06	0.74 ± 0.01	0.77 ± 0.03
Trace	0.57 ± 0.05	0.98 ± 0.00	0.87 ± 0.01	0.83 ± 0.06	0.78 ± 0.03	0.60 ± 0.12	0.96 ± 0.02
TwoLeadECG	0.85 ± 0.02	0.98 ± 0.00	0.96 ± 0.01	0.88 ± 0.03	0.91 ± 0.03	0.90 ± 0.02	0.93 ± 0.02
TwoPatterns	0.45 ± 0.02	1.00 ± 0.00	0.87 ± 0.03	0.52 ± 0.03	0.49 ± 0.06	1.00 ± 0.00	1.00 ± 0.00
UMD	0.54 ± 0.03	0.82 ± 0.02	0.83 ± 0.04	0.60 ± 0.07	0.61 ± 0.04	0.58 ± 0.09	0.65 ± 0.04
UWaveGestureLibraryAll	0.60 ± 0.20	0.97 ± 0.00	0.92 ± 0.01	0.80 ± 0.14	0.50 ± 0.14	0.70 ± 0.24	0.84 ± 0.20
UWaveGestureLibraryX	0.69 ± 0.02	0.82 ± 0.00	0.76 ± 0.00	0.78 ± 0.00	0.67 ± 0.02	0.80 ± 0.01	0.80 ± 0.02
UWaveGestureLibraryY	0.70 ± 0.01	0.76 ± 0.01	0.71 ± 0.01	0.73 ± 0.01	0.61 ± 0.01	0.74 ± 0.01	0.75 ± 0.00
UWaveGestureLibraryZ	0.68 ± 0.02	0.77 ± 0.00	0.72 ± 0.01	0.74 ± 0.01	0.58 ± 0.02	0.73 ± 0.01	0.76 ± 0.01
Wafer	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Wine	0.59 ± 0.09	0.83 ± 0.02	0.87 ± 0.05	0.80 ± 0.07	0.64 ± 0.06	0.73 ± 0.11	0.78 ± 0.05
WordSynonyms	0.36 ± 0.01	0.72 ± 0.01	0.62 ± 0.01	0.53 ± 0.01	0.41 ± 0.04	0.51 ± 0.04	0.61 ± 0.02
Worms	0.43 ± 0.01	0.43 ± 0.00	0.42 ± 0.02	0.43 ± 0.00	0.44 ± 0.04	0.43 ± 0.01	0.48 ± 0.03
WormsTwoClass	0.50 ± 0.01	0.64 ± 0.02	0.54 ± 0.03	0.49 ± 0.05	0.52 ± 0.02	0.54 ± 0.03	0.52 ± 0.03
Yoga	0.84 ± 0.02	0.97 ± 0.00	0.89 ± 0.01	0.93 ± 0.01	0.80 ± 0.03	0.91 ± 0.01	0.95 ± 0.01