Pattern Recognition Letters 155 (2022) 54–61

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Mixing up contrastive learning: Self-supervised representation learning for time series

Kristoffer Wickstrøm[a,*], Michael Kampffmeyer[a,b], Karl Øyvind Mikalsen[a,c], Robert Jenssen[a,b]

[a] UiT Machine Learning Group at the Department of Physics and Technology, UiT the Arctic University of Norway, Tromsø NO-9037, Norway
[b] Norwegian Computing Center, Department SAMBA, P.O. Box 114 Blindern, NO-0314 Oslo, Norway
[c] Department of Gastrointestinal Surgery, University Hospital of North Norway (UNN), Tromsø, Norway

ARTICLE INFO

Article history:
Received 9 April 2021
Revised 5 February 2022
Accepted 12 February 2022
Available online 14 February 2022

Edited by Dr. P. Sarkar

Keywords:
Time series
Self-supervised learning
Contrastive learning
Mixup
Transfer learning

ABSTRACT

The lack of labeled data is a key challenge for learning useful representation from time series data. However, an unsupervised representation framework that is capable of producing high quality representations could be of great value. It is key to enabling transfer learning, which is especially beneficial for medical applications, where there is an abundance of data but labeling is costly and time consuming. We propose an unsupervised contrastive learning framework that is motivated from the perspective of label smoothing. The proposed approach uses a novel contrastive loss that naturally exploits a data augmentation scheme in which new samples are generated by mixing two data samples with a mixing component. The task in the proposed framework is to predict the mixing component, which is utilized as soft targets in the loss function. Experiments demonstrate the framework's superior performance compared to other representation learning approaches on both univariate and multivariate time series and illustrate its benefits for transfer learning for clinical time series.

1. Introduction

Learning a useful representation of time series without labels is a challenging task. Nevertheless, time series are a typical data type in numerous domains where the lack of labeled data is a common challenge. Particularly in the medical domain there can often be an abundance of data but labeling can be costly and challenging [1]. Learning useful representations from unlabeled data would be of great benefit in such scenarios. In particular, it could enable transfer learning for clinical time series. Transfer learning is the practice of transferring knowledge from a source domain to a target domain [2]. Such a technique enables researchers to exploit large unlabeled datasets to train more robust and precise systems on small labeled datasets.

Learning useful representations is an active area of research in machine learning [3,4], with encouraging results in recent works on image representation learning [5–7]. Many of such recent works have used contrastive learning for learning useful features, and these works exploits prior information about noise invariances in the image data. However, time series data constitute a highly heterogeneous data source, and invariances can differ completely between different datasets.

Contrastive learning is a type of self-supervised representation learning where the task is to discriminate between different views of the sample, where the different views are created through data augmentation that exploit prior information about the structure in the data. Data augmentation is typically performed by injecting noise into the data. Recent advances in contrastive learning have been particularly prominent for image data, as there exists a wide range of applicable augmentation schemes [5,8] that are suitable for natural images. On the other hand, data augmentation for time series based on the injection of noise can be more challenging because of the heterogeneous nature of time series data and the lack of generally applicable augmentations.

This paper introduces a novel self-supervised learning framework that naturally exploits a recent data augmentation scheme called mixup [9]. The mixup data augmentation scheme creates an augmented sample through a convex combination of two data points and a mixing component. Such an approach allows for a natural generation of new data points, as augmented samples are

[*] Corresponding author.
E-mail address: kristoffer.k.wickstrom@uit.no (K. Wickstrøm).

https://doi.org/10.1016/j.patrec.2022.02.007
0167-8655/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

generated through a combination of samples from the data distribution. In the proposed framework, the task is to predict the strength of the mixing component based on the two data points and the augmented sample, which is motivated by recent research on label smoothing [10]. Label smoothing refers to the concept of adding noise to the labels, such that the targets are no longer hard 0 and 1 targets, but soft targets in the range between 0 and 1. This has been shown to increase performance and reduce overconfidence in deep learning-based approaches [10]. The proposed framework shows encouraging results when evaluated on the UCR [11] and UEA [12] databases and compared to a number of baselines. Furthermore, we show how the proposed method can be used to enable transfer learning for clinical time series. Experiments illustrate that self-supervised pre-training can increase both performance and convergence speed for deep learning-based classification of clinical time series.

Our contributions are:

1. A novel contrastive learning framework that is motivated through the concept of label smoothing and is based on predicting the amount of mixing between data points.
2. An extensive evaluation of the proposed method with comparison to a number of baselines.
3. We show how the proposed method enables transfer learning clinical time series, which leads to an increase in performance when classifying echocardiograms.

2. Mixup contrastive learning

We outline the proposed framework for contrastive representation learning of time series. We propose a new contrastive loss that naturally exploits the information from the data augmentation procedure. Before we present our new contrastive learning framework, we introduce some notation. Our presentation will be based on univariate time series (UTS), but is also extended to multivariate time series (MTS) in the experiments. Let a UTS, x, be defined as a sequence of real numbers ordered in time, $x = \{x(t) \in \mathbb{R} | t = 1, 2, \dots, T\}$, where t denotes each time step and T denotes the length of the UTS. Vectorial data will be denoted in lowercase bold $\mathbf{x}$.

A common approach to contrastive learning is to use a neural network-based encoder to transform the data into a new representation [5]. The encoder is trained by passing different augmentations of the same sample through the encoder and a projection head, before applying a contrastive loss. The goal of contrastive learning is to embed similar samples in close proximity by exploiting the invariances in the data. After training, the task dependent projection head is discarded and the encoder is kept for down-stream tasks.

The data augmentation scheme used to create different views of the same sample is crucial for learning a useful representation. However, care must be taken when determining the set of transformations to apply. The potential invariances of time series are rarely known in advance, and incautious application can result in a representation where unalike samples are embedded in close proximity [13]. For instance, a transformation like rotation that is common to apply for natural images can completely change the nature of a time series by changing the trend of the data.

In this work, we opt for a data augmentation scheme based on creating new samples through convex combinations of training examples referred to as mixup [9]. Given two time series x_i and x_j drawn randomly from our training data, an augmented training example can be constructed as follows:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j. \tag{1}$$

Here, $\lambda \in [0, 1]$ is a mixing parameter that determines the contribution of each time series in the new sample, where $\lambda \sim$ Beta(α, α) and $\alpha \in (0, \infty)$. The distribution of λ for different values of α is illustrated in Fig. 1. The choice of this augmentation scheme is motivated by avoiding the need to tune a noise parameter based on specific datasets but instead automatically generating data samples based on the specific dataset. Moreover, the information in the mixing parameter λ can be exploited to produce a novel contrastive loss that is described in the following section. In a nutshell, the proposed framework is based on transforming the task from predicting hard 0 and 1 targets to soft targets λ and $1 - \lambda$. This is motivated by recent research on label smoothing that has shown how such regularization can lead to increased performance and less overconfidence in deep learning [10].

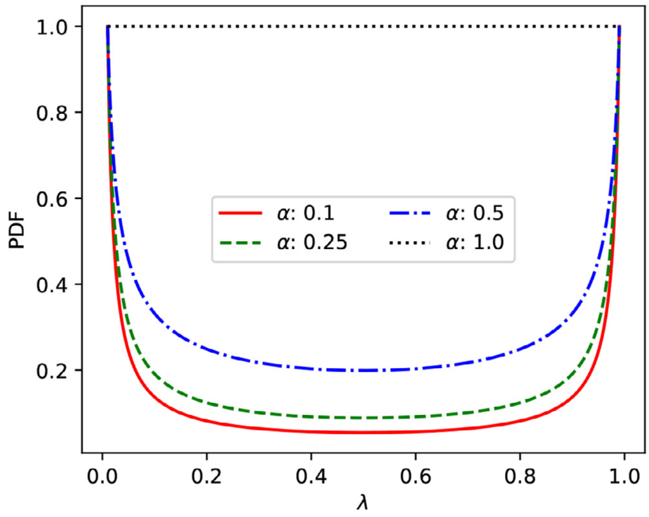

Fig. 1. The beta distribution for different values of α. As α approaches 1 the distribution tends towards a uniform distribution. Larger α results in more mixing of samples.

2.1. A novel contrastive loss for unsupervised representation learning of time series

We propose a new contrastive loss function that naturally exploits the information from the mixing parameter λ. At each training iteration, a new λ is drawn randomly from a beta distribution, and two minibatches of size N, $\{x_1^{(1)}, \dots, x_N^{(1)}\}$ and $\{x_1^{(2)}, \dots, x_N^{(2)}\}$, are drawn randomly from the training data. Applying Eq. (1), the two minibatches are used to create a new minibatch of augmented samples, $\{\tilde{x}_1, \dots, \tilde{x}_N\}$. All three minibatches are passed through the encoder, $f(\cdot)$, that transforms the data into a new representation, $\{\mathbf{h}_1^{(1)}, \dots, \mathbf{h}_N^{(1)}\}$, $\{\mathbf{h}_1^{(2)}, \dots, \mathbf{h}_N^{(2)}\}$, and $\{\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_N\}$, which can be used for down-stream tasks. Next, the new representations are again transformed into a task-dependent representation, $\{\mathbf{z}_1^{(1)}, \dots, \mathbf{z}_N^{(1)}\}$, $\{\mathbf{z}_1^{(2)}, \dots, \mathbf{z}_N^{(2)}\}$, and $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_N\}$, by the projection head, $g(\cdot)$, where the contrastive loss is applied. The framework is illustrated in Fig. 2. Using this notation, our proposed contrastive loss for a single instance is applied on the representation produced by the projection head and is defined as:

$$l_i = -\lambda \log \frac{\exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_i^{(1)})}{\tau})}{\sum\limits_{k=1}^{N} \left(\exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_k^{(1)})}{\tau}) + \exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_k^{(2)})}{\tau}) \right)}$$
$$- (1 - \lambda) \log \frac{\exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_i^{(2)})}{\tau})}{\sum\limits_{k=1}^{N} \left(\exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_k^{(1)})}{\tau}) + \exp(\frac{D_C(\tilde{\mathbf{z}}_i, \mathbf{z}_k^{(2)})}{\tau}) \right)},$$

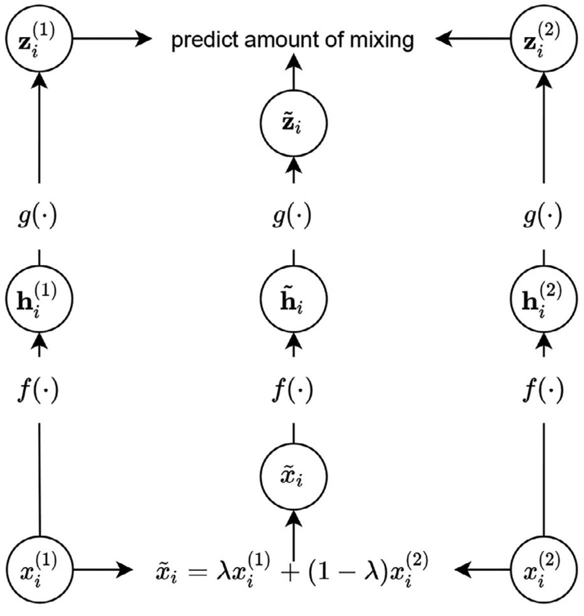

Fig. 2. The proposed framework. Two minibatches are sampled randomly from the data and combined using Eq. (1). All samples are passed though an encoder $f(\cdot)$ resulting in a representation that can be used for down-stream tasks. Next, this representation is transformed using a projection head $g(\cdot)$ into a representation where the proposed contrastive loss is applied.

where $D_C(\cdot)$ denotes the cosine similarity and τ denotes a temperature parameter, as in recent works on contrastive learning [5]. The loss will be referred to as the MNT-Xent loss (the mixup normalized temperature-scaled cross entropy loss). The proposed loss changes the task from identifying the positive pair of samples, as in standard contrastive learning, to predicting the amount of mixing. Moreover, neural networks are known to be overly confident in predictions far from the training data [14], but the proposed loss will discourage overconfidence since the model is tasked with predicting the mixing factor instead of a hard 0 or 1 decision.

3. Experiments and results

We evaluate the proposed framework on an extensive number of both UTS and MTS datasets, and compare against well known baselines. Also, we demonstrate how the proposed methodology enables transfer learning in clinical time series.

3.1. Evaluating quality of representation

A common approach for evaluating the usefulness of an unsupervised contrastive learning framework is training a simple classifier on the learned representation [15,16]. We use a 1-nearest-neighbor (1NN) classifier to evaluate quality of different representations, as suggested by Dau et al. [11]. This is motivated by the simplicity of the 1NN classifier, which requires no training and minimal hyperparameter tuning. Furthermore, the 1NN classifier is highly dependent on the representation to achieve good performance and is therefore a good indicator of the quality of the learned representation. The proposed methodology, referred to as mixup contrastive learning (MCL), is evaluated on the UCR archive [11], which consists of 128 UTS datasets, and the UEA archive [12], which consists of 30 MTS datasets. We compare with several baselines that span different types of time series representations:

- Handcrafted features (HC): Extract the maximum, minimum, variance and mean value of each time series. This is an elementary and well-known approach that will act as a simple baseline.
- Raw input features (ED): Using the raw time series as input without any alterations. This will demonstrate if it is beneficial to transform the time series.
- Autoencoder features (AE): A deep learning-based baseline using an autoencoder framework. We use the same network as with the proposed method, and a mirrored encoder for the decoder. The model is trained using a mean squared error reconstruction loss for 250 epochs. Autoencoder-based learning of features from time series with a reconstruction loss is a typical approach in the literature [17,18]
- Contrastive learning features (CL): A deep learning-based baseline based on the widely used SimCLR framework [5]. We use the same network as with the proposed method, but with different data augmentation and the standard contrastive loss of [5] instead of the mixup contrastive loss. We consider two data augmentation techniques, gaussian noise with a variance of 0.25 (CL ($\sigma = 0.25$)) and dropout noise with a dropout rate of 0.25 (CL ($\rho = 0.25$)). These noise parameters represent an average amount of noise suitable for most datasets.

We use the fully convolutional network (FCN) proposed by Wang et al. [19] as an encoder f for all contrastive learning approaches reported in this work. The FCN consists of three convolutional layers, each followed by batch normalization [20] and a rectified linear unit activation function, and an adaptive average pooling layer. The convolutional layers consist of 128, 256, and 128 filters from first to third layer. This choice is motivated by the FCN's strong performance on a number of time series benchmark tasks Wang et al. [19] and its simplicity. Specifically, the encoder representation will be the output of the average pooling layer. For the projection head g, we use a two-layer neural network with 128 neurons in each layer and separated by a rectified linear unit non-linearity, inspired by Chen et al. [5]. All models are optimized using the ADAM optimizer [21] for 1000 epochs, with the temperature parameter τ is set to 0.5 as suggested by Chen et al. [5], and the α parameter set to 0.2 as suggested by Zhang et al. [9]. Statistical significance is determined using a pairwise t-test, where bold numbers indicate significance at a significance level of 0.05. The accuracy and ranking of the learned features (AE, CL and MCL) are based on the average across 5 training runs at the last epoch.

Table 1 displays the results of the evaluation of the quality of the representation obtained through different representation learning approaches. Results on individual datasets are displayed in the Appendix. Table 1 shows that the simple HC baseline results in poor performance, even compared to no transformation of input (ED). Furthermore, the learned CL feature baseline gives comparable results to the ED features, while the AE features give a slight improvement over the ED features. However, the learned features based on the proposed framework gives the best performance on both the univariate and the multivariate datasets. Fig. 3 shows a per dataset accuracy comparison of the proposed method with all baselines. Each point in Fig. 3 represents the accuracy on one dataset from the UCR and UEA databases, with the baseline along the vertical axis and the MCL along the horizontal axis. The diagonal line indicates where two methods perform equally. Points above this line indicates that the baselines gives better performance and points below this line indicates that the MCL gives better performance. Fig. 3 clearly shows that the majority of the points lie below the diagonal line, which illustrates the superior performance of the proposed method. Lastly, Fig. 4 shows a boxplot of the accuracy across all datasets in the UCR and UEA databases.

Table 1
Accuracy and ranking of a 1NN classifier on different representations averaged over all datasets. Results show that the representation obtained from the proposed method results in better performance across all metrics.

	UCR		UEA	
Features	Avg accuracy	Avg ranking	Avg accuracy	Avg ranking
HC	0.520	1.63	0.560	2.80
ED	0.686	3.86	0.585	3.56
AE	0.702	4.00	0.587	3.56
CL ($\sigma = 0.25$)	0.666	3.41	0.573	3.16
CL ($\rho = 0.25$)	0.660	2.99	0.570	3.06
MCL	**0.759**	**4.81**	**0.627**	**4.26**

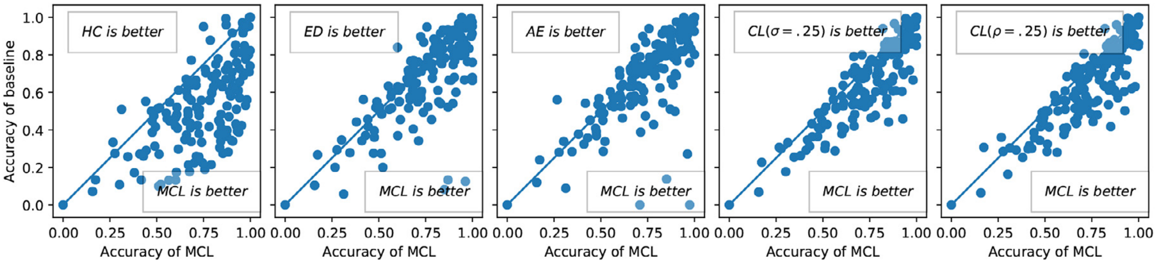

Fig. 3. Accuracy on each dataset from the UCR and UEA databases. Each point represents the accuracy on one dataset, with the baseline along the vertical axis and the MCL along the horizontal axis. The diagonal line indicates where two methods have similar performance. Points above this line indicates that the baselines gives better performance and points below this line indicates that the MCL gives better performance. The figure shows that the proposed method provides superior performance to the baselines, as the majority of the points lie below the diagonal line.

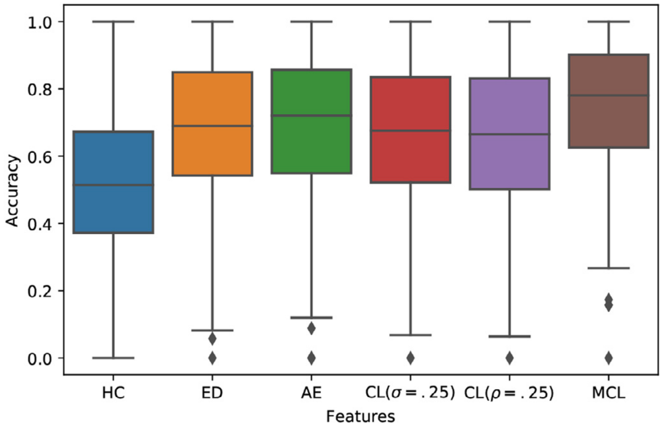

Fig. 4. Boxplot of accuracy across all datasets of the UCR and UEA databases. Figure shows that the proposed framework produces representations that yield better performance at a more consistent rate. The whiskers indicated the lower and upper quartiles, with outliers indicated through black dots.

The figure corroborates Table 1, and illustrates that the proposed method outperforms all other baselines.

3.2. Transfer learning for clinical time series

We perform transfer learning for classification of echocardiograms (ECGs) datasets with limited amount of training data, which is a typical scenario for many clinical time series datasets. First, we train an encoder using the proposed contrastive learning framework on a pretext task where a larger amount of data is available. We consider different domains for the pretext task, but with a similar amount of data. The pretext task datasets are the Syntehetic Control (Synthetic), Swedish Leaf (Dissimilar), and ECG5000 (Similar), all obtained from the UCR archive. Next, we use the weights of the encoder to initialize the weights of a supervised model, in this case the FCN, and train the model using the standard procedure. Additionally, a baseline is included where the weights are randomly initialized using He normal initialization [22].

The results of the transfer learning experiments are presented in Table 2. Using the pretrained weights obtained through the proposed contrastive learning framework leads to improved performance on most datasets. For the ECG200, the random initialization gives the highest performance. This might be a results of the ECG200 having the most training samples of the four datasets. Furthermore, Fig. 5 shows how the accuracy evolves during training, and demonstrates how using pretrained weights can lead to faster convergence and increased performance compared to random initialization. Also note that the models with weights pretrained on the similar and dissimilar domain displays a degree of overfitting after 50 epochs. At this point in the training, the loss has begun to saturate. Therefore, we believe that this overfitting might be a result of the model being to fitted to the pretext task, which hurts the performance for the down-stream task. Such challenges could be addressed through techniques such as early stopping [23] or heavier regularization, which we consider a direction for future research.

Next, the results in Table 2 indicate that the domain of the pretext task is important for the quality of the pretrained weights. Surprisingly, a pretext task from a dissimilar domain results in comparable results as a similar domain. It is natural to assume that a pretext task within a similar domain would be beneficial, but it is important to also consider the complexity of the data in the pretext task. In this case, the Swedish Leaf dataset is more complex as it has more classes and a more erratic nature compared to the periodic ECG5000 dataset. This might result in the

Table 2
Accuracy on test data of ECG datasets with different initialization of the encoder weights. The number of training samples in each dataset is denoted by *N*. Results show how using the weights trained through the proposed contrastive framework can increase performance, particularly when the number of training samples is small.

Pretraining	ECGFiveDays $N = 23$	TwoLeadECG $N = 23$	ECG200 $N = 100$	CinCECGTorso $N = 40$
Random	0.989 ± 0.002	0.967 ± 0.004	**0.874** ± **0.008**	0.616 ± 0.033
Synthetic	0.997 ± 0.001	0.985 ± 0.002	0.866 ± 0.014	0.644 ± 0.019
Dissimilar	0.999 ± 0.001	**0.987** ± **0.004**	0.868 ± 0.010	**0.715** ± **0.024**
Similar	0.998 ± 0.001	**0.995** ± **0.001**	0.842 ± 0.012	**0.696** ± **0.022**

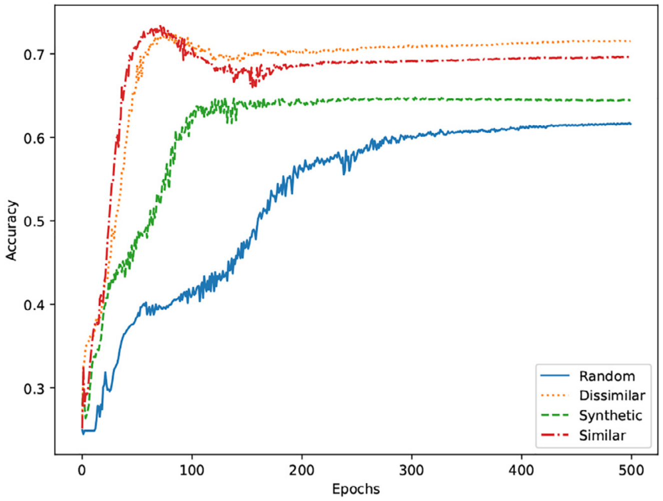

Fig. 5. Accuracy of a FCN with different encoder initialization on the CinCECGTorso test data. Scores are averaged over 5 independent training runs. The figure shows that the pretrained weights using the proposed framework leads to faster convergence and increased performance.

encoder learning filters that can process more complicated data and generalize better to different tasks. Moreover, using the encoder trained on synthetic data also increased performance on some datasets, which indicates that useful information can be extracted even from generated data. This can be helpful for tasks with little data and no pretext task, as you can generate data and learn filters to initialize the model which might lead to a better representation.

4. Discussion and conclusion

In this work, we have focused on contrastive learning of time series representations through the injection of noise, motivated by the recent success of contrastive learning on image data. However, a different line of research for contrastive learning of time series representations is using temporal information to discriminate between samples. Most recently, Franceschi et al. [24] achieved promising results by combining a convolutional neural network encoder with a novel triplet loss, where temporal information was used to perform negative-sampling. Banville et al. [25] proposed a self-supervised learning approach where an informative representation was obtained by predicting whether time windows are sampled from the same temporal context or not. Hyvarinen and Morioka [26] proposed a time-contrastive learning principle that uses the non-stationary structure of the data to learn a representation where optimal discrimination of time segments is encouraged, and demonstrated how the time-contrastive learning could be related to nonlinear independent component analysis. Hyvärinen et al. [27] also proposed a generalized contrastive learning framework with connections to nonlinear independent component analysis. Exploiting temporal information can be beneficial when such information is discriminative but can also encounter challenges when faced with periodic data, where noise-based approaches might succeed. We envision that our noise-based approached can be combined with temporal-based contrastive learning to reap the benefits of both approaches, and consider such a combination a promising line of future research. Lastly, a possible direction to improve the transfer learning part of our work is to include memory-based merging of features, as proposed by Ding et al. [28]. Such an approach could allow for samples from the source and target domain to be merged and potentially increase performance.

This paper introduced a novel self-supervised framework for time series representation learning. The framework exploits a recent augmentation technique called miuxp, in which new samples are generated through combinations of data points. The proposed framework was evaluated on numerous datasets with encouraging results. Furthermore, we demonstrated how the proposed framework enables transfer learning for clinical time series with good results. We believe that our proposed framework can be a useful approach for time series representation learning.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was financially supported by the Research Council of Norway (RCN), through its Centre for Research-based Innovation funding scheme (Visual Intelligence, grant no. 309439), and Consortium Partners. The work was further partially funded by RCN FRIPRO grant no. 315029, RCN IKTPLUSS grant no. 303514, and the UiT Thematic Initiative “Data-Driven Health Technology”.

Appendix A. Results on individual datasets

Tables A.3–A.5 displays the accuracy of all methods evaluated in the article on all datasets of the UCR and UEA databases, respectively. For the learning-based methods (AE, CL, and MCL), the scores represent the average accuracy across five independent training runs. Results for all 5 training runs and ranks on individual datasets are available at https://github.com/Wickstrom/MixupContrastiveLearning along with code.

Table A.3
Accuracy of a 1NN classifier on UCR datasets starting with the letters A-R. For AE, CLG, CLD, and MCL, the results are the average across 5 different training runs. Datasets from A-R.

Dataset	HC	ED	AE	$CL(\rho = 0.25)$	$CL(\sigma = 0.25)$	MCL
ACSF1	0.58	0.54	0.50	0.76	0.75	0.90
Adiac	0.41	0.61	0.58	0.54	0.55	0.68
AllGestureWiimoteX	0.37	0.52	0.58	0.53	0.51	0.66
AllGestureWiimoteY	0.38	0.57	0.55	0.58	0.55	0.73
AllGestureWiimoteZ	0.30	0.45	0.46	0.41	0.41	0.62
ArrowHead	0.65	0.80	0.81	0.61	0.63	0.82
BME	0.56	0.83	0.83	0.67	0.63	0.98
Beef	0.50	0.67	0.67	0.42	0.43	0.67
BeetleFly	0.95	0.75	0.75	0.76	0.72	0.75
BirdChicken	0.70	0.55	0.73	0.94	0.94	0.82
CBF	0.66	0.85	0.95	0.99	1.00	0.94
Car	0.42	0.73	0.73	0.48	0.48	0.78
Chinatown	0.51	0.95	0.88	0.71	0.87	0.93
ChlorineConcentration	0.45	0.65	0.53	0.46	0.45	0.66
CinCECGTorso	0.50	0.90	0.90	0.52	0.52	0.72
Coffee	0.57	1.00	1.00	0.96	0.96	0.94
Computers	0.58	0.58	0.58	0.60	0.61	0.67
CricketX	0.26	0.58	0.56	0.55	0.53	0.71
CricketY	0.18	0.57	0.55	0.54	0.47	0.68
CricketZ	0.26	0.59	0.56	0.60	0.52	0.72
Crop	0.48	0.71	0.70	0.54	0.56	0.73
DiatomSizeReduction	0.99	0.93	0.94	0.86	0.84	0.87
DistalPhalanxOutlineAgeGroup	0.61	0.63	0.64	0.69	0.68	0.62
DistalPhalanxOutlineCorrect	0.67	0.72	0.73	0.71	0.71	0.66
DistalPhalanxTW	0.49	0.63	0.60	0.60	0.58	0.55
DodgerLoopDay	0.31	0.55	0.57	0.41	0.44	0.49
DodgerLoopGame	0.59	0.88	0.85	0.72	0.69	0.79
DodgerLoopWeekend	0.81	0.99	0.99	0.91	0.87	0.95
ECG200	0.72	0.88	0.90	0.80	0.77	0.87
ECG5000	0.85	0.92	0.93	0.92	0.92	0.92
ECGFiveDays	0.73	0.80	0.82	0.86	0.91	0.94
EOGHorizontalSignal	0.29	0.42	0.45	0.36	0.36	0.44
EOGVerticalSignal	0.17	0.44	0.38	0.26	0.25	0.38
Earthquakes	0.64	0.71	0.70	0.63	0.64	0.69
ElectricDevices	0.41	0.55	0.56	0.54	0.55	0.61
EthanolLevel	0.27	0.27	0.29	0.32	0.32	0.51
FaceAll	0.21	0.71	0.69	0.66	0.67	0.79
FaceFour	0.41	0.78	0.79	0.63	0.81	0.85
FacesUCR	0.34	0.77	0.77	0.76	0.79	0.91
FiftyWords	0.13	0.63	0.59	0.38	0.37	0.60
Fish	0.27	0.78	0.80	0.65	0.64	0.85
FordA	0.53	0.67	0.67	0.81	0.79	0.88
FordB	0.51	0.61	0.61	0.68	0.68	0.73
FreezerRegularTrain	0.94	0.80	0.88	0.90	0.90	0.96
FreezerSmallTrain	0.88	0.68	0.70	0.69	0.69	0.79
Fungi	0.39	0.82	0.82	0.70	0.69	0.93
GestureMidAirD1	0.13	0.58	0.58	0.29	0.28	0.56
GestureMidAirD2	0.10	0.49	0.45	0.30	0.28	0.51
GestureMidAirD3	0.11	0.35	0.34	0.18	0.16	0.30
GesturePebbleZ1	0.34	0.73	0.71	0.69	0.68	0.68
GesturePebbleZ2	0.35	0.67	0.63	0.58	0.58	0.65
GunPoint	0.74	0.91	0.93	0.85	0.85	1.00
GunPointAgeSpan	0.71	0.90	0.98	0.95	0.96	0.99
GunPointMaleVersusFemale	0.82	0.97	1.00	0.98	0.96	1.00
GunPointOldVersusYoung	1.00	0.95	1.00	1.00	1.00	1.00
Ham	0.45	0.60	0.52	0.52	0.56	0.57
HandOutlines	0.63	0.86	0.86	0.70	0.69	0.80
Haptics	0.26	0.37	0.36	0.31	0.30	0.43
Herring	0.55	0.52	0.56	0.51	0.50	0.57
HouseTwenty	0.47	0.66	0.64	0.87	0.77	0.89
InlineSkate	0.27	0.34	0.32	0.29	0.30	0.41
InsectEPGRegularTrain	1.00	0.68	1.00	1.00	1.00	1.00
InsectEPGSmallTrain	1.00	0.66	1.00	1.00	1.00	1.00
InsectWingbeatSound	0.13	0.56	0.54	0.35	0.32	0.42
ItalyPowerDemand	0.61	0.96	0.94	0.91	0.93	0.94
LargeKitchenAppliances	0.56	0.49	0.47	0.73	0.74	0.74
Lightning2	0.56	0.75	0.77	0.76	0.75	0.78
Lightning7	0.42	0.58	0.62	0.55	0.58	0.72
Mallat	0.38	0.91	0.92	0.85	0.84	0.83
Meat	0.48	0.93	0.93	0.87	0.86	0.84
MedicalImages	0.43	0.68	0.64	0.62	0.61	0.68
MelbournePedestrian	0.42	0.85	0.94	0.85	0.85	0.93
MiddlePhalanxOutlineAgeGroup	0.47	0.52	0.54	0.49	0.49	0.48
MiddlePhalanxOutlineCorrect	0.64	0.77	0.75	0.73	0.73	0.67

(*continued on next page*)

Table A.3 (*continued*)

Dataset	HC	ED	AE	$CL(\rho = 0.25)$	$CL(\sigma = 0.25)$	MCL
MiddlePhalanxTW	0.42	0.51	0.49	0.45	0.47	0.49
MixedShapesRegularTrain	0.43	0.90	0.90	0.60	0.60	0.92
MixedShapesSmallTrain	0.44	0.84	0.83	0.54	0.53	0.84
MoteStrain	0.74	0.88	0.84	0.85	0.84	0.88
NonInvasiveFetalECGThorax1	0.34	0.83	0.82	0.57	0.61	0.84
NonInvasiveFetalECGThorax2	0.40	0.88	0.87	0.68	0.69	0.88
OSULeaf	0.37	0.52	0.51	0.66	0.61	0.87
OliveOil	0.33	0.87	0.87	0.71	0.70	0.74
PLAID	0.70	0.52	0.72	0.56	0.54	0.76
PhalangesOutlinesCorrect	0.64	0.76	0.74	0.71	0.73	0.71
Phoneme	0.09	0.11	0.12	0.19	0.19	0.22
PickupGestureWiimoteZ	0.48	0.56	0.74	0.55	0.56	0.67
PigAirwayPressure	0.51	0.06	0.09	0.33	0.30	0.31
PigArtPressure	0.69	0.12	0.27	0.72	0.62	0.96
PigCVP	0.63	0.08	0.14	0.44	0.42	0.85
Plane	0.80	0.96	0.97	0.97	0.96	0.99
PowerCons	0.92	0.93	0.98	0.94	0.93	0.90
ProximalPhalanxOutlineAgeGroup	0.72	0.79	0.79	0.78	0.78	0.76
ProximalPhalanxOutlineCorrect	0.68	0.81	0.78	0.75	0.77	0.80
ProximalPhalanxTW	0.60	0.71	0.71	0.70	0.68	0.66
RefrigerationDevices	0.45	0.39	0.39	0.49	0.46	0.48
Rock	0.42	0.84	0.72	0.42	0.41	0.60

Table A.4
Accuracy of a 1NN classifier on UCR datasets starting with the letters S-Y. For AE, CLG, CLD, and MCL, the results are the average across 5 different training runs.

Dataset	HC	ED	AE	$CL(\rho = 0.25)$	$CL(\sigma = 0.25)$	MCL
ScreenType	0.39	0.36	0.37	0.41	0.42	0.48
SemgHandGenderCh2	0.78	0.76	0.91	0.79	0.77	0.83
SemgHandMovementCh2	0.47	0.37	0.69	0.58	0.56	0.60
SemgHandSubjectCh2	0.56	0.40	0.84	0.68	0.66	0.68
ShakeGestureWiimoteZ	0.62	0.60	0.81	0.84	0.83	0.91
ShapeletSim	0.45	0.54	0.54	0.79	0.75	0.83
ShapesAll	0.30	0.75	0.73	0.64	0.63	0.84
SmallKitchenAppliances	0.52	0.34	0.39	0.67	0.67	0.71
SmoothSubspace	0.81	0.91	0.81	0.87	0.87	0.92
SonyAIBORobotSurface1	0.64	0.70	0.67	0.79	0.74	0.67
SonyAIBORobotSurface2	0.65	0.86	0.85	0.83	0.85	0.83
StarLightCurves	0.85	0.85	0.86	0.86	0.85	0.97
Strawberry	0.70	0.95	0.94	0.86	0.86	0.96
SwedishLeaf	0.34	0.79	0.79	0.86	0.84	0.90
Symbols	0.39	0.90	0.89	0.83	0.77	0.94
SyntheticControl	0.42	0.88	0.93	0.98	0.99	0.95
ToeSegmentation1	0.63	0.68	0.69	0.80	0.78	0.90
ToeSegmentation2	0.71	0.81	0.79	0.85	0.79	0.90
Trace	1.00	0.76	0.80	0.89	0.86	1.00
TwoLeadECG	0.67	0.75	0.70	0.75	0.72	0.90
TwoPatterns	0.28	0.91	0.92	0.97	0.96	0.88
UMD	0.94	0.76	0.76	0.86	0.85	0.97
UWaveGestureLibraryAll	0.19	0.95	0.94	0.46	0.44	0.76
UWaveGestureLibraryX	0.22	0.74	0.73	0.54	0.49	0.74
UWaveGestureLibraryY	0.21	0.66	0.63	0.48	0.44	0.67
UWaveGestureLibraryZ	0.22	0.65	0.64	0.53	0.50	0.70
Wafer	0.95	1.00	0.99	0.98	0.98	0.99
Wine	0.48	0.61	0.63	0.64	0.61	0.61
WordSynonyms	0.17	0.62	0.58	0.39	0.39	0.61
Worms	0.56	0.45	0.43	0.60	0.56	0.78
WormsTwoClass	0.65	0.61	0.62	0.64	0.64	0.82
Yoga	0.60	0.83	0.80	0.75	0.76	0.79

Table A.5
Accuracy of a 1NN classifier on all UEA datasets. For AE, CLG, CLD, and MCL, the results are the average across 5 different training runs.

Dataset	HC	ED	AE	$CL(\rho = 0.25)$	$CL(\sigma = 0.25)$	MCL
ArticularyWordRecognition	0.78	0.97	0.97	0.87	0.90	0.97
AtrialFibrillation	0.13	0.27	0.24	0.23	0.31	0.17
BasicMotions	1.00	0.68	0.97	1.00	0.98	1.00
CharacterTrajectories	0.82	0.96	0.94	0.94	0.90	0.98
Cricket	0.92	0.94	0.90	0.91	0.94	0.96
DuckDuckGeese	0.50	0.28	0.41	0.36	0.34	0.47
ERing	0.67	0.13	0.91	0.83	0.83	0.87
EigenWorms	0.66	0.55	0.00	0.61	0.62	0.71
Epilepsy	0.97	0.67	0.83	0.93	0.93	0.96
EthanolConcentration	0.27	0.29	0.28	0.30	0.29	0.28
FaceDetection	0.51	0.52	0.52	0.50	0.50	0.50
FingerMovements	0.52	0.55	0.52	0.53	0.51	0.61
HandMovementDirection	0.26	0.28	0.28	0.25	0.29	0.35
Handwriting	0.11	0.20	0.34	0.43	0.42	0.52
Heartbeat	0.65	0.62	0.70	0.69	0.70	0.68
InsectWingbeat	0.00	0.00	0.00	0.00	0.00	0.00
JapaneseVowels	0.96	0.92	0.92	0.87	0.88	0.87
LSST	0.55	0.46	0.45	0.43	0.39	0.44
Libras	0.61	0.83	0.78	0.61	0.57	0.89
MotorImagery	0.46	0.51	0.53	0.56	0.55	0.56
NATOPS	0.66	0.85	0.84	0.76	0.73	0.82
PEMS-SF	0.66	0.70	0.79	0.80	0.80	0.71
PenDigits	0.53	0.97	0.00	0.86	0.87	0.97
Phoneme	0.07	0.10	0.12	0.07	0.06	0.16
RacketSports	0.75	0.87	0.79	0.80	0.79	0.82
SelfRegulationSCP1	0.77	0.77	0.78	0.72	0.72	0.68
SelfRegulationSCP2	0.52	0.48	0.49	0.52	0.50	0.52
SpokenArabicDigits	0.76	0.97	0.94	0.46	0.50	0.93
StandWalkJump	0.33	0.20	0.56	0.31	0.36	0.27
UWaveGestureLibrary	0.37	0.88	0.83	0.61	0.60	0.87

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2022.02.007.

References

[1] T. Ching, D.S. Himmelstein, B.K. Beaulieu-Jones, et al., Opportunities and obstacles for deep learning in biology and medicine, J. R. Soc. Interface (2018) 20170387, doi:10.1098/rsif.2017.0387.

[2] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.

[3] Y. Bengio, A.C. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[4] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: a survey, IEEE Trans. Pattern Anal. Mach. Intell. (2019) 1, doi:10.1109/TPAMI.2020.2992393.

[5] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, 2020, pp. 1597–1607.

[6] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[7] J.-B. Grill, F. Strub, F. Altché, et al., Bootstrap your own latent: a new approach to self-supervised learning, in: Advances in Neural Information Processing Systems, 2020, pp. 21271–21284.

[8] C. Shorten, T. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 1–48.

[9] H. Zhang, M. Cissé, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[10] R. Müller, S. Kornblith, G.E. Hinton, When does label smoothing help? in: Advances in Neural Information Processing Systems, 2019, pp. 4694–4703.

[11] H.A. Dau, A.J. Bagnall, K. Kamgar, et al., The UCR time series archive, 2018. http://arxiv.org/abs/1810.07758.

[12] A.J. Bagnall, H.A. Dau, J. Lines, et al., The UEA multivariate time series classification archive, 2018, CoRR abs/1811.00075(2018).

[13] J. Oh, J. Wang, J. Wiens, Learning to exploit invariances in clinical time-series data using sequence transformer networks, in: Proceedings of Machine Learning Research, Volume 85, Palo Alto, California, 2018, pp. 332–347.

[14] M. Hein, M. Andriushchenko, J. Bitterwolf, Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem, in: Conference on Computer Vision and Pattern Recognition, 2019, pp. 41–50.

[15] R. Zhang, P. Isola, A.A. Efros, Split-brain autoencoders: unsupervised learning by cross-channel prediction, in: Conference on Computer Vision and Pattern Recognition, 2017, pp. 645–654.

[16] M. Caron, P. Bojanowski, A. Joulin, et al., Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 139–156.

[17] L. Zhu, H. Fan, Y. Luo, M. Xu, Y. Yang, Temporal cross-layer correlation mining for action recognition, IEEE Trans. Multimed. (2021) 1, doi:10.1109/TMM.2021.3057503.

[18] T. Kieu, B. Yang, C. Guo, C.S. Jensen, Outlier detection for time series with recurrent autoencoder ensembles, in: International Joint Conference on Artificial Intelligence, 2019, pp. 2725–2732.

[19] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: a strong baseline, in: Proceedings of the International Joint Conference on Neural Networks, 2017, pp. 1578–1585.

[20] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[21] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2014.

[22] K. He, X. Zhang, S. Ren, et al., Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[23] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural Comput. 7 (1995) 219–269.

[24] J.-Y. Franceschi, A. Dieuleveut, M. Jaggi, Unsupervised scalable representation learning for multivariate time series, in: Advances in Neural Information Processing Systems, 2019, pp. 4650–4661.

[25] H.J. Banville, I. Albuquerque, A. Hyvarinen, Self-supervised representation learning from electroencephalography signals, in: International Workshop on Machine Learning for Signal Processing, 2019, pp. 1–6.

[26] A. Hyvarinen, H. Morioka, Unsupervised feature extraction by time-contrastive learning and nonlinear ICA, in: Advances in Neural Information Processing Systems, 2016, pp. 3765–3773.

[27] A. Hyvärinen, H. Sasaki, R. Turner, Nonlinear ICA using auxiliary variables and generalized contrastive learning, in: The 22nd International Conference on Artificial Intelligence and Statistics, 2019, pp. 859–868.

[28] Y. Ding, H. Fan, M. Xu, Y. Yang, Adaptive exploration for unsupervised person re-identification, ACM Trans. Multimed. Comput., Commun., Appl. 16 (1) (2020) 1–19, doi:10.1145/3369393.