applied sciences

MDPI

Article

BIMO: Bootstrap Inter–Intra Modality at Once Unsupervised Learning for Multivariate Time Series

Seongsil Heo, Sungsik Kim and Jaekoo Lee *

College of Computer Science, Kookmin University, Seoul 02707, Republic of Korea
* Correspondence: jaekoo@kookmin.ac.kr

Abstract: It is difficult to learn meaningful representations of time-series data since they are sparsely labeled and unpredictable. Hence, we propose bootstrap inter–intra modality at once (BIMO), an unsupervised representation learning method based on time series. Unlike previous works, the proposed BIMO method learns both inter-sample and intra-temporal modality representations simultaneously without negative pairs. BIMO comprises a main network and two auxiliary networks, namely inter-auxiliary and intra-auxiliary networks. The main network is trained to learn inter–intra modality representations sequentially by regulating the use of each auxiliary network dynamically. Thus, BIMO thoroughly learns inter–intra modality representations simultaneously. The experimental results demonstrate that the proposed BIMO method outperforms the state-of-the-art unsupervised methods and achieves comparable performance to existing supervised methods.

Keywords: biomedical and health informatics; deep learning; unsupervised learning; time series; modality

1. Introduction

The volume of time-series data is rapidly growing with various applications in a wide variety of domains. Considerable developments have been noted in several fields, such as signal processing and machine learning [1–4]. Recently, deep learning models for time-series data have demonstrated remarkable performances [5–10].

Most of these models adopt a supervised learning approach, which has to collect a massive amount of data with high-quality data annotation. Therefore, we explore a time-series unsupervised learning approach to tackle data acquisition problems.

Unsupervised learning attempts to identify meaningful generalized properties from unlabeled data. Unsupervised learning has recently attracted significant attention, particularly in computer vision. The contrastive learning method is prominent among various unsupervised learning methods [11–17]. In addition, recent attempts have been made to remove negative pairs, which is a problem in the contrastive learning method [15,18].

However, unsupervised learning with time-series data has not been studied as extensively in computer vision, and some challenges remain in existing methods. Most time-series data are unpredictable and nonstationary [19,20], thus existing methods are limited with regard to extracting meaningful generalized properties.

Unsupervised learning-based time-series models can be broadly categorized into two approaches, those that learn inter-sample modality representations [21,22] and those that learn inter-temporal modality representations [23,24]. Inter-sample modality representation derives relationships between two samples. In contrast, intra-temporal modality representation derives features according to time within the same samples.

Most previous studies focused on training specific modality representations. In addition, the contrastive learning method requires attentive treatment while collecting proper negative pairs.

check for updates

Citation: Heo, S.; Kim, S.; Lee, J. BIMO: Bootstrap Inter–Intra Modality at Once Unsupervised Learning for Multivariate Time Series. *Appl. Sci.* **2024**, *14*, 3825. https://doi.org/10.3390/app14093825

Academic Editor: Yang Kuang

Received: 2 April 2024
Accepted: 25 April 2024
Published: 30 April 2024

Therefore, in this paper, we propose the Bootstrap Inter–Intra Modality at Once (BIMO) method, which is an unsupervised learning method for multivariate time series that simultaneously explores inter–intra modality representations without negative pairs. The proposed BIMO method comprises three neural networks: the main network and two auxiliary networks (i.e.g., inter-auxiliary and intra-auxiliary networks). These three networks interact and learn from each other.

From given raw time-series data, two transformed samples are generated using an augmentation strategy: (1) the input to the main network and (2) the input to the inter-auxiliary network. The input of the main network generates another sample, which is the input of the intra-auxiliary network, using a subsampling strategy. The main network simultaneously predicts the representation of the two samples generated from the two auxiliary networks. The proposed BIMO method learns the complementary properties in both modalities efficiently and simultaneously by adjusting the weight of each auxiliary network dynamically.

We measured the performance of the learned representation with various datasets to validate the generalizability of the proposed method. Here, we used univariate UCR datasets [25], which are well-known time-series datasets. We showed that the proposed BIMO method is universal, comparable to state-of-the-art (SOTA) time-series supervised methods, and superior to previous time-series unsupervised methods.

We also evaluated the performance of the proposed method on multivariate UEA datasets [26]. Here, we found that the proposed BIMO method is suitable for representation learning with multivariate time-series data. We then used a real-world wearable stress and affection detection (WESAD) dataset to demonstrate the noise robustness of the proposed BIMO method.

Our primary contributions are summarized as follows. (1) We propose a unsupervised learning-based time-series simple method that trains the main network using two auxiliary networks while exploring inter–intra modality representations simultaneously. (2) We remove the constraints for negative pairs from contrastive learning-based time-series data analysis. (3) We present various comprehensive analyses to extract robust features, considering inter–intra modality representations, from the unsupervised learning perspective of time-series data. (4) We utilize various datasets to verify that the proposed BIMO method is universal, robust against noise, and outperforms contemporary SOTA methods.

2. Materials and Methods

BIMO's goal is to be easily used in downstream tasks by discovering the most significant modalities for representation learning in all domains of time-series data. This study was inspired by existing work on SOTA contrastive learning-based unsupervised learning methods [15,23,27].

As shown in Figure 1, the proposed BIMO method consists of the main network and two auxiliary networks. The main network consists of an encoder f_θ, a projector g_θ, and a predictor q_θ, and each auxiliary network comprises an encoder and a projector. The main network learns to have a similar distribution between two values p'_ξ, p''_λ from the respective projectors of the two auxiliary networks and a value $q_\theta(p_\theta)$ from the predictor of the main network.

It is a significant issue to simultaneously learn both inter- and intra-modality representations. We trained the proposed BIMO method to learn inter–intra modality representation efficiently and stably based on the fundamental concept, i.e., high-level features comprise low-level and intermediate-level features [28].

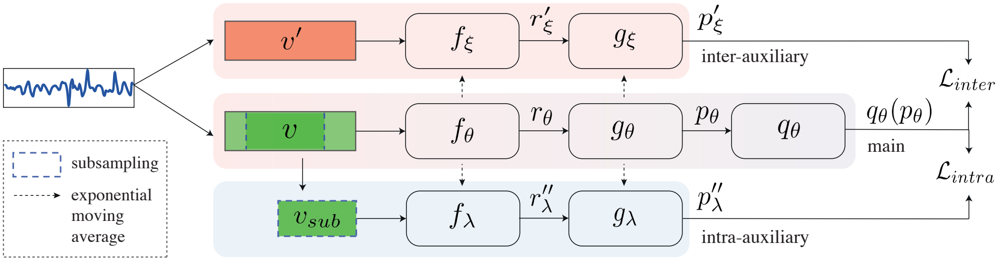


Figure 1. BIMO's architecture: f, g, and q represent the encoder, projector, and predictor, respectively.

An overview of the training process in the proposed BIMO method is given in Algorithm 1. The complexity of the proposed BIMO method is $O(4N)$, while the complexity of USRL as a existing SOTA method is at least $O(18N)$.

While training, we first used a hard constraint in the inter-auxiliary network to learn sufficient low-level coarse information, i.e., the time characteristics within samples, from the intra-auxiliary network. As the number of epochs increased, we gradually applied a hard constraint to the intra-auxiliary network and not to the inter-auxiliary network. Therefore, the proposed BIMO method sufficiently learns fine-grained features, i.e., the correlation between two augmented samples, from the inter-auxiliary network.

Therefore, BIMO learns low-level features sufficiently at the initial training step, and gradually learns high-level features.

Algorithm 1 BIMO's training procedure

```
Input: Time series set $X = \{x_n\}_{n=1}^{N}$ , Number epochs $M$
Output: Trained $f_\theta$
 1: Initialization $f_\theta, g_\theta, q_\theta, f_\lambda, g_\lambda, f_\xi, g_\xi$ ← initialize weights
 2: $m \leftarrow 1$
 3: repeat
 4:     for $n = 1$ to $N$ with $s_n = size(x_n)$ do
 5:         generate $v \triangleq t(x)$, $v' \triangleq t'(x)$ from different augmentation $t \sim \tau, t' \sim \tau$
 6:         extract $s_{v_{sub}} = size(v)$ in $[\![1, s_n]\!]$
 7:         extract $v_{sub}$ among subseries of $v$ of length $s_{v_{sub}}$
 8:         $r_\theta \leftarrow f_\theta(v)$, $p_\theta \leftarrow g_\theta(r_\theta)$, $q_\theta(p_\theta)$
 9:         $r'_\xi \leftarrow f_\xi(v')$, $p'_\xi \leftarrow g_\xi(r'_\xi)$
10:         $r''_\lambda \leftarrow f_\lambda(v_{sub})$, $p''_\lambda \leftarrow g_\lambda(r''_\lambda)$
11:         $\mathcal{L}_{inter} \leftarrow \|\bar{q}_\theta(p_\theta) - \bar{p}'_\xi\|_2^2$
12:         $\mathcal{L}_{intra} \leftarrow \|\bar{q}_\theta(p_\theta) - \bar{p}''_\lambda\|_2^2$
13:         $\mathcal{L}_{BIMO} \leftarrow (1 - \frac{1}{m})(\mathcal{L}_{inter}) + \frac{1}{m}(\mathcal{L}_{intra})$
14:         $f_\theta, g_\theta, q_\theta$ ← update weights using $\mathcal{L}_{BIMO}$
15:         $f_\lambda, g_\lambda, f_\xi, g_\xi$ ← update weights using moving exponential average
16:     end for
17:     $m \leftarrow m + 1$
18: until m = M
```

2.1. BIMO's Components

Given time-series data, $X = \{x_n\}_{n=1}^{N}$, where N is the volume of data, which comprises a token $x_n = (x_{n,1}, \ldots, x_{n,T})$, T ordered real values.

The proposed BIMO method consists of three networks, and each network uses a set of weights: θ, ξ, and λ.

A sample x generates two augmented views $v \triangleq t(x)$ and $v' \triangleq t'(x)$, which apply two augmentations $t \sim \tau$ and $t' \sim \tau$ ($line$5). For the augmentation strategy, we employ a magnitude domain augmentation method, which transforms the values of the time-series data, and a time domain augmentation method, which transforms the time-series data sequence. Here, v is the input of the main network, and v' is the input of the inter-auxiliary network; v_{sub}, which is the input of the intra-auxiliary network, is subsampled from v ($lines$6–7), and M is the number of epochs.

2.2. Training Details

We first forward the three generated samples ($lines$8–10). The main and inter-auxiliary networks learn representation through the generated samples from the same time-series data in different augmentation approaches. Therefore, the proposed BIMO method learns to have similar distributions between $q_\theta(p_\theta)$ from the predictor of the main network and p'_ξ from the projector of the inter-auxiliary network ($line$11).

The inputs of the intra-auxiliary network are subsamples from the input of the main network. Hence, the samples are highly likely to have similar distributions since they are in similar periods. The proposed BIMO method also learns to have similar distribution between $q_\theta(p_\theta)$ from the predictor of the main network and p''_λ from the projector of the intra-auxiliary network ($line$12).

First, we train the main network with the intra-auxiliary network in a high ratio and the inter-auxiliary network in a low ratio to learn the low-level coarse information at an initial time based on the fundamental principles of deep learning [28] ($line$13). Then, we gradually decrease the ratio of the intra-auxiliary network and increase the ratio of the inter-auxiliary network in every epoch. We only minimize the loss function with a single weight, θ, in each training step ($line$14). The other weights (i.e., ξ and λ) prevent network collapse using slowly moving average methods, which is $\tau\xi \leftarrow (1-\tau)\theta$ ($line$15).

The output of the main network is $q_\theta(g_\theta(f_\theta(v))) \triangleq q_\theta(p_\theta)$, the output of the inter-auxiliary networks is $g_\xi(f_\xi(v')) \triangleq p'_\xi$, and the output of the intra-auxiliary networks is $g_\lambda(f_\lambda(v_{sub})) \triangleq p''_\lambda$. Each output $q_\theta(p_\theta)$, p'_ξ, and p''_λ applies ℓ_2-normalization and becomes $\bar{q_\theta}(p_\theta) \triangleq q_\theta(p_\theta)/\|q_\theta(p_\theta)\|_2$, $\bar{p'_\xi} \triangleq p'_\xi/\|p'_\xi|_2$, and $\bar{p''_\lambda} \triangleq p''_\lambda/\|p''_\lambda|_2$, respectively. Thus, the training objective aims to minimize the differences between $q_\theta(p_\theta)$ and p'_ξ as well as $q_\theta(p_\theta)$ and p''_λ. Losses are defined as follows:

$$\mathcal{L}_{inter} \triangleq \|\bar{q_\theta}(p_\theta) - \bar{p'_\xi}\|_2^2 = 2 - 2 \cdot \frac{< q_\theta(p_\theta), p'_\xi >}{\|q_\theta(p_\theta)\|_2 \cdot \|p'_\xi\|_2} \tag{1}$$

$$\mathcal{L}_{intra} \triangleq \|\bar{q_\theta}(p_\theta) - \bar{p''_\lambda}\|_2^2 = 2 - 2 \cdot \frac{< q_\theta(p_\theta), p''_\lambda >}{\|q_\theta(p_\theta)\|_2 \cdot \|p''_\lambda\|_2} \tag{2}$$

$$\mathcal{L}_{BIMO} = (1 - \frac{1}{m})(\mathcal{L}_{inter} + \tilde{\mathcal{L}}_{inter}) + \frac{1}{m}(\mathcal{L}_{intra} + \tilde{\mathcal{L}}_{intra}) \tag{3}$$

Equations (1) and (2) represent the inter and intra losses, respectively, and Equation (3) represents the total loss. $\tilde{\mathcal{L}}_{inter}$ and $\tilde{\mathcal{L}}_{intra}$ in Equation (3) exchange between v and v' to symmetrize the losses, where m denotes a training epoch.

2.3. Architecture and Optimization

Time-series data have to accommodate varying lengths and be efficient in terms of time and memory, as such data are often updated in real time. Thus, we used a dilated causal convolution network [23,29,30] as a backbone to fulfil the requirements.

The dilated causal convolution network comprises 20 layers, each of which exponentially increases the dilation parameter: 2^i for the i-th layer. We employ an adaptive max-pooling layer as the last layer to squeeze the temporal dimension and output a vector of a fixed size. Here, representation r is projected into a multilayer perceptron (MLP), g_θ, comprising two layers, and projection p is forwarded into another MLP, q_θ, which has the same structure as g_θ. We used the output dimensions of 512 and 320 for the first and second layers of the MLPs, respectively. For the auxiliary networks, we began with the exponential moving average parameter $\tau_{base} = 0.996$ and increased it to 1 during training.

3. Results and Discussion

We performed classification tasks to evaluate the proposed BIMO method's validity in representation learning. We used typical time-series datasets: univariate UCR datasets [25] and multivariate UEA datasets [26]. We also used a public wearable dataset, the WESAD dataset [31], to validate BIMO's robustness against noisy data. The encoder was trained on an unlabeled training set, and the learned encoder was used to perform a classification task. In addition, we trained a simple single-layer linear classifier on a labeled training set [32–34].

3.1. Implementation

Sample Generation: Time-series augmentation can be divided into magnitude-based and time-based methods. In this study, we used the time-series augmentation set t and t', which comprises magnitude-based magnitude warping and scaling methods, and time-based time-slicing and time-warping methods [35,36].

The time-series subsampling strategy is based on the literature [23]. We randomly extracted a part of the samples by selecting the length and starting point. We selected different lengths and starting points for each epoch and trained them with various lengths of subsamples to learn a sufficient inter-temporal modality representation.

Encoder Selection: Time-series data should comprise temporal orders, which are required to consider temporal information, accommodate unequal lengths, and be efficient in terms of both time and memory. Note that deep convolutional neural networks (CNNs) do not consider temporal information and are difficult to apply to data of various lengths. Long short-term memory (LSTM) is inefficient in terms of time and memory. Thus, we used exponentially dilated causal convolutions to handle these issues [23,29,30].

To verify the conformity of our encoder selection, we measured the classification performance on the UCR datasets using dilated causal convolutions, ResNet, and a two-layer LSTM encoder. Each model outperformed the other two on 65%, 35%, and 5% of the first 20 UCR datasets, respectively. This result confirmed that the encoder with dilated causal convolution was the most suitable for the proposed BIMO method. The accuracy results are detailed in Table 1.

Table 1. Accuracy scores depending on encoder type with first 15 UCR datasets. Encoder type includes dilated convolution (DConv.), LSTM, and ResNet. Bold text represents the best accuracy.

Dataset	DConv. (BIMO)	ResNet	LSTM
Adiac	**0.760**	0.482	0.342
ArrowHead	**0.814**	0.763	0.388
Beef	**0.800**	0.625	0.313
BeetleFly	**0.850**	0.688	0.750
BirdChicken	**0.900**	0.750	0.563
Car	**0.917**	0.688	0.417
CBF	**0.998**	0.992	0.401
ChlorineConcentration	0.635	**0.731**	0.534
CinCECGTorso	**0.757**	0.629	0.283
Coffee	**1.000**	**1.000**	0.625
Computers	0.681	**0.729**	0.571
CricketX	**0.750**	0.651	0.107
CricketY	**0.716**	0.628	0.216
CricketZ	**0.758**	0.378	0.102
DiatomSizeReduction	**0.977**	0.911	0.336
DistalPhalanxOutlineAgeGroup	0.743	**0.820**	0.523
DistalPhalanxOutlineCorrect	0.786	**0.809**	0.581
DistalPhalanxTW	0.684	**0.688**	0.422
Earthquakes	0.765	0.727	**0.767**
ECG200	0.900	**0.906**	0.698

3.2. Univariate Time Series

We validated the proposed BIMO method's performance using the 85 initially released UCR datasets, which are representative univariate time-series datasets [25]. (1) We compared the BIMO method's performance to that of existing SOTA unsupervised models, (2) with the existing SOTA supervised models, and (3) compared the performance depending on combinations of the auxiliary networks.

Overall Performance: In terms of performance, we compared the proposed BIMO method with unsupervised models for time series, i.e., USRL (which utilizes triplet loss) [23], DTW (which employs a kernel-based estimation method) [37], and RWS (which uses a similarity matrix) [38], as shown in Table 2.

Table 2. Accuracy scores of BIMO, SOTA unsupervised models (USRL, and DTW), and supervised models (BOSS, PF, ResNet, HIVE-COTE and ITime). Bold text represents the best accuracy among the unsupervised models; * denotes the best accuracy, while underlined text represents the second-best accuracy among all models.

Dataset	**Unsupervised**			**Supervised**			
	BIMO	**USRL**	**DTW**	**BOSS**	**PF**	**HIVE-COTE**	**ITime**
Adiac	**0.760**	0.716	0.604	0.765	0.734	0.811	0.836 *
ArrowHead	0.814	**0.829**	0.703	0.834	0.875 *	0.863	0.829
Beef	**0.800**	0.700	0.633	0.800	0.720	0.933 *	0.700
BeetleFly	0.850	**0.900**	0.700	0.900	0.875	0.950 *	0.850
BirdChicken	**0.900**	0.800	0.750	0.950 *	0.865	0.867	0.950 *
Car	**0.917 ***	0.817	0.733	0.833	0.847	0.867	0.900
CBF	**0.998**	0.994	0.997	0.998	0.993	0.999 *	0.998
ChlCon	0.635	**0.782**	0.648	0.661	0.634	0.712	0.875 *
CinCECGTorso	**0.757**	0.740	0.651	0.887	0.934	0.996 *	0.851
Coffee	**1.000 ***	**1.000 ***	**1.000 ***	1.000 *	1.000 *	1.000 *	1.000 *
Computers	0.681	0.628	**0.700**	0.756	0.644	0.760	0.812 *
CricketX	0.750	**0.777**	0.754	0.736	0.802	0.823	0.867 *
CricketY	0.716	**0.767**	0.744	0.754	0.794	0.849	0.851 *
CricketZ	0.758	**0.764**	0.754	0.746	0.801	0.831	0.859 *
DiaSizRed	0.977	**0.993 ***	0.967	0.931	0.966	0.941	0.931
DisPhaOutAgeGroup	0.743	0.734	**0.770 ***	0.748	0.731	0.763	0.727
DisPhaxOutCorrect	**0.786**	0.768	0.717	0.728	0.793	0.772	0.794 *
DistalPhalanxTW	**0.684 ***	0.676	0.590	0.676	0.660	0.683	0.676
Earthquakes	**0.765 ***	0.748	0.719	0.748	0.754	0.748	0.741
ECG200	**0.900**	**0.900**	0.770	0.870	0.909	0.850	0.910 *
ECG5000	**0.940**	0.936	0.924	0.941	0.937	0.946 *	0.941
ECGFiveDays	**1.000 ***	**1.000 ***	0.768	1.000 *	0.849	1.000 *	1.000 *
ElectricDevices	0.632	**0.732**	0.602	0.799 *	0.706	0.770	0.723
FaceAll	**0.839**	0.802	0.808	0.782	0.894 *	0.803	0.804
FaceFour	0.841	**0.875**	0.830	1.000 *	0.974	0.955	0.966
FacesUCR	**0.948**	0.918	0.905	0.957	0.946	0.963	0.973 *
FiftyWords	**0.783**	0.780	0.690	0.705	0.831	0.809	0.842 *
Fish	**0.959**	0.880	0.823	0.989*	0.935	0.989 *	0.983
FordA	0.850	**0.935**	0.555	0.930	0.855	0.964 *	0.948
FordB	0.714	**0.810**	0.620	0.711	0.715	0.823	0.937 *
GunPoint	**1.000 ***	0.993	0.907	1.000 *	0.997	1.000 *	1.000 *
Ham	**0.740 ***	0.695	0.467	0.667	0.660	0.667	0.714
HandOutlines	**0.924**	0.922	0.881	0.903	0.921	0.932	0.960 *
Haptics	**0.510**	0.455	0.377	0.461	0.445	0.519	0.568 *
Herring	**0.703 ***	0.578	0.531	0.547	0.580	0.688	0.703 *
InlineSkate	0.372	**0.447**	0.384	0.516	0.542 *	0.500	0.486
InsWinbeatSound	**0.630**	0.623	0.355	0.523	0.619	0.655 *	0.635
ItalyPowerDemand	**0.963**	0.925	0.950	0.909	0.967	0.963	0.968 *
LarKitAppliances	**0.866**	0.848	0.795	0.765	0.782	0.864	0.907 *
Lightning2	0.883	**0.918 ***	0.869	0.836	0.866	0.820	0.803

Table 2. *Cont.*

Dataset	Unsupervised			Supervised			
	BIMO	**USRL**	**DTW**	**BOSS**	**PF**	**HIVE-COTE**	**ITime**
Lightning7	**0.819**	0.795	0.726	0.685	0.822 *	0.740	0.808
Mallat	0.956	**0.964** *	0.934	0.938	0.958	0.962	0.963
Meat	**1.000** *	0.950	0.933	0.900	0.933	0.933	0.950
MedicalImages	0.730	**0.784**	0.737	0.718	0.758	0.778	0.799 *
MidPhaOutAgeGroup	0.618	**0.656** *	0.500	0.545	0.562	0.597	0.533
MidPhaOutCorrect	**0.826**	0.814	0.698	0.780	0.836 *	0.832	0.835
MiddlePhalanxTW	0.566	**0.610** *	0.506	0.545	0.529	0.571	0.513
MoteStrain	**0.871**	**0.871**	0.835	0.879	0.902	0.933 *	0.903
NonInvFetECGTho1	**0.923**	0.910	0.790	0.838	0.906	0.930	0.962 *
NonInvFetECGTho2	**0.929**	0.927	0.865	0.901	0.940	0.945	0.967 *
OliveOil	**0.964** *	0.900	0.833	0.867	0.867	0.900	0.867
OSULeaf	0.729	**0.831**	0.591	0.955	0.827	0.979 *	0.934
PhaOutCorrect	**0.801**	**0.801**	0.728	0.772	0.824	0.807	0.854 *
Phoneme	0.263	**0.289**	0.228	0.265	0.320	0.382 *	0.335
Plane	**1.000** *	0.990	**1.000** *	1.000 *	1.000 *	1.000 *	1.000 *
ProPhaOutAgeGroup	**0.863** *	0.854	0.805	0.834	0.846	0.859	0.854
ProPhaOutCorrect	**0.878**	0.859	0.784	0.849	0.873	0.880	0.931 *
ProximalPhalanxTW	0.814	**0.824** *	0.761	0.800	0.779	0.815	0.776
RefrigerationDevices	**0.524**	0.517	0.464	0.499	0.532	0.557 *	0.509
ScreenType	**0.446**	0.413	0.397	0.464	0.455	0.589 *	0.576
ShapeletSim	0.694	**0.817**	0.650	1.000 *	0.776	1.000 *	0.989
ShapesAll	0.667	**0.875**	0.768	0.908	0.886	0.905	0.925 *
SmaKitAppliances	**0.790**	0.715	0.643	0.725	0.744	0.853 *	0.779
SonAIBORobSur1	**0.967** *	0.897	0.725	0.632	0.846	0.765	0.884
SonAIBORobSur2	0.858	**0.934**	0.831	0.859	0.896	0.928	0.953 *
StarLightCurves	**0.970**	0.965	0.907	0.978	0.981	0.982 *	0.979
Strawberry	**0.962**	0.946	0.941	0.976	0.968	0.970	0.984 *
SwedishLeaf	0.929	**0.931**	0.792	0.922	0.947	0.954	0.971 *
Symbols	0.960	**0.965**	0.950	0.967	0.962	0.974	0.982 *
SyntheticControl	0.900	0.983	**0.993**	0.967	0.995	0.997 *	0.997 *
ToeSeg1	0.917	**0.952**	0.772	0.939	0.925	0.982 *	0.969
ToeSeg2	**0.891**	0.885	0.838	0.962 *	0.862	0.954	0.939
Trace	**1.000** *	**1.000** *	**1.000** *	1.000 *	1.000 *	1.000 *	1.000 *
TwoLeadECG	0.996	**0.997** *	0.905	0.981	0.989	0.996	0.996
TwoPatterns	**1.000** *	**1.000** *	**1.000** *	0.993	1.000 *	1.000 *	1.000 *
UWavGesLibAll	**0.958**	0.941	0.892	0.939	0.972 *	0.968	0.955
UWavGesLibX	0.802	**0.811**	0.728	0.762	0.829	0.840 *	0.825
UWavGesLibY	0.712	**0.735**	0.634	0.685	0.762	0.765	0.769 *
UWavGesLibZ	0.742	**0.759**	0.658	0.695	0.764	0.783 *	0.770
Wafer	**0.996**	0.993	0.980	0.995	0.996	0.999 *	0.999 *
Wine	0.808	**0.870** *	0.574	0.741	0.569	0.778	0.667
WordSynonyms	0.701	**0.704**	0.649	0.638	0.779 *	0.738	0.756
Worms	0.684	**0.714**	0.584	0.558	0.718	0.558	0.805 *
WormsTwoClass	**0.842** *	0.818	0.623	0.831	0.784	0.779	0.792
Yoga	0.807	**0.878**	0.837	0.918 *	0.879	0.918 *	0.906

We also compared BIMO with supervised models, i.e., PF (which uses a decision tree ensemble) [39], BOSS (which employs a dictionary-based classifier) [5], InceptionTime (ITime) [7], and HIVE-COTE (which uses ensemble methods) [8]. As shown in Figure 2, we compared performance based on the average rank according to the accuracy results on the UCR datasets. All accuracy results are detailed in Table 2.

For the unsupervised models, the proposed BIMO method obtained the best rank scores: 3.71, 3.91, and 6.11 for BIMO, USRL, and DTW, respectively. For the supervised models, BIMO showed the third-highest score: 2.41, 2.52, 3.71, 3.73, and 3.91 for HIVE-COTE, ITime, BIMO, BOSS, and PF, respectively. These results demonstrate that BIMO is superior to existing SOTA unsupervised models and comparable to well-known supervised models.

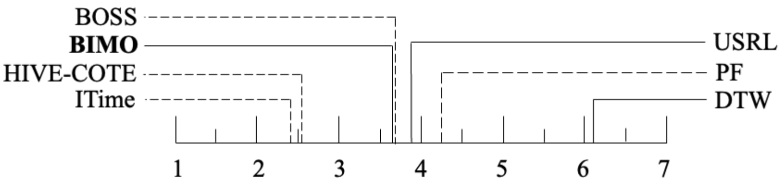


Figure 2. Average rank diagram of BIMO, existing SOTA unsupervised models (USRL, DTW), and supervised models (PF, BOSS, HIVE-COTE, ITime) for the UCR datasets. The average rank means the average of the top ranking results of a model. The black lines indicate an unsupervised models, and dotted lines represent supervised models.

Inter–Intra Modality Representation Ablation: We compared performance depending on the combination of auxiliary networks based on the average rank according to the accuracy results on the UCR datasets. We used a single auxiliary network, e.g., an inter-auxiliary or intra-auxiliary network, and multiple auxiliary networks, e.g., inter-auxiliary and intra-auxiliary networks. As shown in Table 3, we compared the performance in terms of the average rank score. More detailed overall accuracy results are shown in Table 4.

Table 3. Average rank comparison depending on the combination of auxiliary networks: a single auxiliary network (*Inter* or *Intra*) and multiple auxiliary networks (*Inter and Intra*, *Inter* ↦ *Intra*, *Intra* ↦ *Inter*). Bold text represents the best rank score.

Single		**Plural**		
Inter	***Intra***	***Inter and Intra***	***Inter* ↦ *Intra***	***Intra* ↦ *Inter***
2.39	3.33	2.87	3.33	**1.90**

Table 4. Accuracy scores depending on the combination of auxiliary networks for the first and recent UCR datasets: using a single auxiliary network (*Inter* or *Intra*) and plural auxiliary networks (*Inter and Intra*, *Inter* ↦ *Intra*, *Intra* ↦ *Inter*). Bold text represents the best accuracy, and the underlined text represents the second-best accuracy.

Dataset	**Single**		**Plural**		
	Inter	***Intra***	***Inter and Intra***	***Inter* ↦ *Intra***	***Intra* ↦ *Inter*** **(BIMO)**
Adiac	**0.778**	0.693	0.729	0.642	0.760
ArrowHead	**0.831**	0.767	0.826	0.785	0.814
Beef	0.750	0.786	0.786	**0.821**	0.800
BeetleFly	0.850	0.850	**0.900**	0.850	0.850
BirdChicken	0.850	**0.900**	0.800	0.797	**0.900**
Car	**0.917**	0.850	0.883	0.983	0.916
CBF	0.990	0.993	0.996	0.986	**0.998**
ChlCon	0.613	0.627	0.597	**0.733**	0.635
CinCECGTorso	0.745	0.737	0.766	**1.000**	0.757
Coffee	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Computers	0.633	0.621	**0.681**	0.625	**0.681**
CricketX	**0.771**	0.629	0.683	0.649	0.750
CricketY	0.696	0.585	0.686	0.652	**0.716**
CricketZ	0.750	0.626	0.706	0.670	**0.758**
DiaSizRed	0.961	0.980	0.964	**0.984**	0.977
DisPhaOutAgeGroup	0.735	0.699	0.735	0.721	**0.786**
DisPhaxOutCorrect	**0.772**	**0.772**	0.761	0.750	0.743
DistalPhalanxTW	**0.721**	0.669	0.669	0.713	0.684
Earthquakes	0.750	0.735	0.735	0.706	**0.765**

Table 4. *Cont.*

Dataset	**Single**		**Plural**		
	Inter	*Intra*	*Inter and Intra*	*Inter ↦ Intra*	*Intra ↦ Inter* **(BIMO)**
ECG200	0.890	0.890	0.880	0.890	**0.900**
ECG5000	0.940	0.940	**0.941**	0.939	0.940
ECGFiveDays	0.991	0.998	0.995	0.997	**1.000**
ElectricDevices	0.606	0.521	0.625	0.585	**0.632**
FaceAll	0.830	0.680	0.771	0.701	**0.839**
FaceFour	0.853	**0.875**	0.841	0.841	0.841
FacesUCR	0.947	0.920	0.922	0.917	**0.948**
FiftyWords	0.774	**0.792**	0.785	0.788	0.783
Fish	**0.959**	0.901	0.948	0.913	**0.959**
FordA	0.867	**0.920**	0.870	0.918	0.850
FordB	0.718	**0.788**	0.756	0.775	0.714
GunPoint	**1.000**	0.986	0.993	0.986	**1.000**
Ham	**1.000**	0.760	0.740	0.712	0.740
HandOutlines	0.921	0.916	0.902	0.913	**0.924**
Haptics	**0.916**	0.494	0.523	0.487	0.510
Herring	0.594	0.688	0.625	**0.703**	**0.703**
InlineSkate	0.352	0.367	0.367	**0.374**	0.372
InsWinbeatSound	0.608	0.598	0.609	0.597	**0.630**
ItalyPowerDemand	0.955	0.954	0.952	**0.963**	**0.963**
LarKitAppliances	**0.871**	0.621	0.863	0.659	0.866
Lightning2	0.767	0.783	0.767	0.717	**0.883**
Lightning7	0.778	0.778	0.764	0.750	**0.819**
Mallat	0.898	0.829	0.920	0.875	**0.956**
Meat	**1.000**	0.950	0.983	0.983	**1.000**
MedicalImages	0.733	0.726	0.730	**0.746**	0.730
MidPhaOutAgeGroup	0.533	0.658	0.618	0.605	**0.826**
MidPhaOutCorrect	0.799	0.792	0.799	**0.823**	0.618
MiddlePhalanxTW	**0.586**	0.559	0.566	0.533	0.566
MoteStrain	0.854	0.851	0.859	0.851	**0.871**
NonInvFetECGTho1	0.916	0.891	0.907	0.894	**0.923**
NonInvFetECGTho2	0.926	0.905	0.920	0.907	**0.929**
OliveOil	**1.000**	0.964	0.964	0.964	0.964
OSULeaf	0.717	0.650	0.696	0.667	**0.729**
PhaOutCorrect	0.780	0.783	0.793	0.770	**0.801**
Phoneme	0.249	0.216	**0.275**	0.220	0.263
Plane	**1.000**	0.990	0.990	**1.000**	**1.000**
ProPhaOutAgeGroup	0.848	0.843	0.814	0.843	**0.878**
ProPhaOutCorrect	0.885	0.882	**0.892**	0.865	0.863
ProximalPhalanxTW	0.789	**0.819**	0.784	0.789	0.814
RefrigerationDevices	0.538	0.556	0.530	**0.559**	0.524
ScreenType	**0.460**	0.454	0.457	0.419	0.446
ShapeletSim	0.583	0.628	0.600	0.639	**0.694**
ShapesAll	0.662	0.647	0.663	0.647	**0.667**
SmaKitAppliances	0.755	0.728	**0.796**	0.726	0.790
SonAIBORobSur1	**0.970**	0.942	0.953	0.960	0.967
SonAIBORobSur2	0.853	0.860	0.843	**0.873**	0.858
StarLightCurves	**0.978**	0.963	0.963	0.955	0.970
Strawberry	0.962	0.943	**0.965**	0.948	0.962
SwedishLeaf	**0.929**	0.925	0.918	0.923	**0.929**
Symbols	0.929	0.935	0.945	0.933	**0.960**

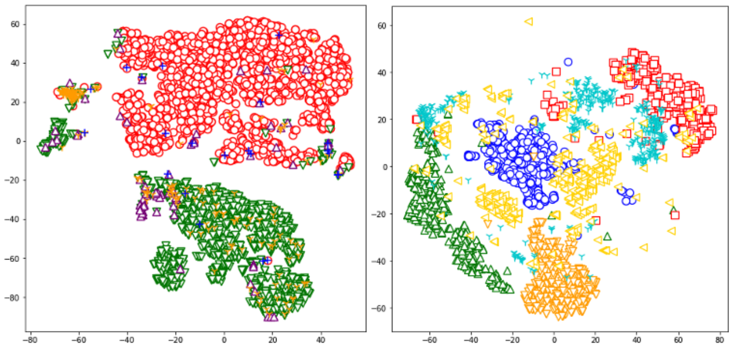
Table 4. *Cont.*

Dataset	Single		Plural		
	Inter	*Intra*	*Inter and Intra*	*Inter* ↦ *Intra*	*Intra* ↦ *Inter* (BIMO)
SyntheticControl	0.873	0.850	0.863	0.873	**0.900**
ToeSeg1	**0.917**	**0.917**	**0.917**	0.886	**0.917**
ToeSeg2	0.883	0.883	0.883	0.883	**0.891**
Trace	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
TwoLeadECG	**0.996**	**0.996**	0.995	**0.996**	**0.996**
TwoPatterns	**1.000**	0.997	**1.000**	0.996	**1.000**
UWavGesLibAll	0.954	0.956	0.951	**0.962**	0.742
UWavGesLibX	0.804	0.799	0.811	0.802	**0.958**
UWavGesLibY	0.726	0.686	0.717	0.690	**0.802**
UWavGesLibZ	0.735	0.719	**0.748**	0.734	0.712
Wafer	**0.996**	0.991	0.995	0.991	**0.996**
Wine	**0.827**	0.731	0.750	0.731	0.808
WordSynonyms	0.682	0.690	0.700	0.662	**0.701**
Worms	0.671	0.618	0.658	0.592	**0.684**
WormsTwoClass	0.803	0.724	0.763	0.711	**0.842**
Yoga	0.807	**0.810**	0.796	0.797	0.807

Given multiple auxiliary networks, we employed the static and dynamic loss functions. During training, the static loss function had an equal ratio of inter-auxiliary and intra-auxiliary networks *(Inter and Intra*). The dynamic loss function had different ratios of the inter- and intra-auxiliary networks for every epoch. Herein, the main network was initially trained with the inter-auxiliary network at a higher ratio than that of the intra-auxiliary network. Then, the ratio of the intra-auxiliary network was increased gradually (*Inter* ↦ *Intra*). In contrast, the main network was trained with the intra-auxiliary network in a higher ratio than that used for the inter-auxiliary network at first; gradually, the ratio of the inter-auxiliary network was increased (*Intra* ↦ *Inter*), which is the training method of BIMO.

As shown in Table 3, the *Intra* ↦ *Inter* method obtained the best rank score. We confirmed that the initial training trained the intra-modality representations sufficiently, which are the relatively low-level features, and then the inter modality representations, which are the relatively high-level features. The proposed dynamic training method made the main network evenly learn both modality representations.

Representation Metric Space: We also validated the performance of representation learning for some UCR datasets using embedding visualization with dimensionality reduction. The results are shown in Figure 3.



(a) ECG5000 (b) UWaveGestureLibraryAll

Figure 3. Visualization of embedded vectors of ECG500 and UWaveGestureLibrary UCR test datasets with dimensionality reduction. Each class marked with different shapes and colors is well differentiable.

3.3. Multivariate Time Series

We validated the performance of BIMO for UEA datasets. Here, we compared the performance of BIMO with USRL and DTW. The accuracy results are shown in Table 5. The BIMO, USRL, and DTW models, respectively, showed the best accuracies for approximately 50%, 32%, and 18% of the datasets. Overall, BIMO's performance is comparable to that of SOTA unsupervised models for multivariate time series.

Table 5. Accuracy of BIMO and SOTA unsupervised methods (USRL, DTW) on UEA datasets. Bold text indicates the best accuracy.

Dataset	BIMO	USRL	DTW
ArticularyWordRecognition	0.830	**0.987**	**0.987**
AtrialFibrillation	**0.417**	0.133	0.200
BasicMotions	**1.000**	**1.000**	0.975
Cricket	0.861	0.986	**1.000**
DuckDuckGeese	**0.688**	0.675	0.600
EigenWorms	0.852	**0.878**	0.618
Epilepsy	0.926	0.957	**0.964**
Ering	**0.922**	0.133	0.133
EthanolConcentration	**0.354**	0.236	0.323
FaceDetection	**0.550**	0.528	0.529
FingerMovements	**0.550**	0.540	0.530
HandMovementDirection	**0.444**	0.270	0.231
Handwriting	0.346	**0.533**	0.286
Heartbeat	**0.740**	0.737	0.717
Libras	0.650	0.867	**0.870**
LSST	0.404	**0.558**	0.551
MotorImagery	**0.600**	0.540	0.500
NATOPS	0.872	**0.944**	0.883
PEMS-SF	**0.733**	0.688	0.711
PenDigits	0.975	**0.983**	0.977
Phoneme	**0.280**	0.246	0.151
RacketSports	0.737	**0.862**	0.803
SelfRegulationSCP1	**0.853**	0.846	0.775
SelfRegulationSCP2	0.550	**0.556**	0.539
StandWalkJump	**0.500**	0.400	0.200
UWaveGestureLibrary	0.819	0.884	**0.903**

3.4. Robustness to Noisy Data

Most real-world time-series data contain some noise. Typically, the photoplethysmogram (PPG) signal, which is also referred to as the blood volume pulse, contains many noises. A PPG signal is simple and highly useful in daily life since it can be easily measured from the wrist. However, it is difficult to apply in an end-to-end deep learning model because it is susceptible to many internal and external noises of the measurement environment [40,41]. Therefore, most existing PPG-based studies have focused on signal processing and feature engineering [4,31,42–44].

In this study, we validated the noise robustness of BIMO, which is an end-to-end deep learning model, using noisy PPG signals. We used a PPG signal from the WESAD dataset [31]. The WESAD dataset is labeled with four emotional states: baseline, stress, amusement, and meditation. We performed a classification task with leave-one-subject-out cross-validation, stress versus nonstress, where nonstress is defined by combining the state baseline and amusement states [31].

We compared the performance with BIMO and existing SOTA supervised learning models for PPG, which is a weak feature engineering method [31] and a strong feature engineering method named OMDP [4]. The weak feature engineering-based method uses a peak detection algorithm, which is computed by simple statistical features. OMDP employs a two-step signal processing method in terms of both time and frequency and an ensemble-based peak detection method; it extracts diverse features from detected peaks.

As a result, we found that BIMO outperformed the supervised learning methods (Table 6), indicating that BIMO is comparable to previous SOTA models. This is a very meaningful result, since BIMO opens up the possibility that unsupervised end-to-end data-driven feature learning is also possible for noisy time-series data.

Table 6. Comparison of accuracy and F1 scores of BIMO and existing models using a PPG signal in the WESAD dataset. Abbreviations: decision tree (DT), random forest (RF), Adaboost (AB), linear discriminant analysis (LDA), k-nearest neighbor (kNN), and feature engineering (FE).

			Accuracy (F1)			
ML Algorithms	**DT**	**RF**	**AB**	**LDA**	**kNN**	**BIMO (Ours)**
weak FE	0.78 (0.81)	0.81 (0.84)	0.81 (0.84)	0.83 (0.86)	0.79 (0.82)	0.87 (0.85)
strong FE (OMDP)	0.87 (0.81)	0.91 (0.87)	0.91 (0.87)	0.97 (0.93)	0.89 (0.89)	

4. Conclusions

We proposed BIMO, which is an unsupervised learning method that is applicable to sparsely labeled and unpredictable time-series data. BIMO learns general features by considering both inter-modality and intra-modality representations simultaneously. In the proposed BIMO method, two auxiliary networks are employed to train the main network, and different ratios of the two auxiliary networks are dynamically applied to learn both modalities efficiently. BIMO demonstrated superior representation learning performance compared to SOTA unsupervised models, and it demonstrated comparable performance to well-known supervised models. In addition, we examined how BIMO is universal and robust to noisy data. The trained encoder of the main network could also be used in many different tasks by fine-tuning the model using simple classifiers.

Author Contributions: Conceptualization, J.L. and S.H.; writing—original draft preparation, J.L. and S.H.; writing—review and editing, J.L. and S.K.; supervision, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation (NRF) grant (No. RS-2023-00212484) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2022-00167194) funded by the Korea government(MSIT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This research was partially supported by the Kookmin University Industry–Academic Cooperation Foundation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bone, D.; Lee, C.C.; Chaspari, T.; Gibson, J.; Narayanan, S. Signal processing and machine learning for mental health research and clinical applications [perspectives]. *IEEE Signal Process. Mag.* **2017**, *34*, 195–196. [CrossRef]
2. Costello, Z.; Martin, H.G. A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data. *NPJ Syst. Biol. Appl.* **2018**, *4*, 19. [CrossRef] [PubMed]
3. Parmezan, A.R.S.; Souza, V.M.; Batista, G.E. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Inf. Sci.* **2019**, *484*, 302–337. [CrossRef]
4. Heo, S.; Kwon, S.; Lee, J. Stress Detection With Single PPG Sensor by Orchestrating Multiple Denoising and Peak-Detecting Methods. *IEEE Access* **2021**, *9*, 47777–47785. [CrossRef]
5. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1578–1585.

6. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
7. Fawaz, H.I.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D.F.; Weber, J.; Webb, G.I.; Idoumghar, L.; Muller, P.A.; Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.* **2020**, *34*, 1936–1962. [CrossRef]
8. Dempster, A.; Petitjean, F.; Webb, G.I. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.* **2020**, *34*, 1454–1495. [CrossRef]
9. Kim, I.; Kim, D.; Kwon, S.; Lee, S.; Lee, J. Fall detection using biometric information based on multi-horizon forecasting. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 1364–1370.
10. Kim, I.; Lim, J.; Lee, J. Human Activity Recognition via Temporal Fusion Contrastive Learning. *IEEE Access* **2024**, *12*, 20854–20866. [CrossRef]
11. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
12. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.
13. Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; Hinton, G. Big self-supervised models are strong semi-supervised learners. *arXiv* **2020**, arXiv:2006.10029.
14. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.
15. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
16. Kim, D.; Yoo, Y.; Park, S.; Kim, J.; Lee, J. Selfreg: Self-supervised contrastive regularization for domain generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9619–9628.
17. Kim, D.; Kim, J.; Lee, J. Inter-domain curriculum learning for domain generalization. *ICT Express* **2022**, *8*, 225–229. [CrossRef]
18. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
19. Tsay, R.S. *Analysis of Financial Time Series*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 543.
20. Cowpertwait, P.S.; Metcalfe, A.V. *Introductory Time Series with R*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
21. Pascual, S.; Ravanelli, M.; Serra, J.; Bonafonte, A.; Bengio, Y. Learning problem-agnostic speech representations from multiple self-supervised tasks. *arXiv* **2019**, arXiv:1904.03416.
22. Sarkar, P.; Etemad, A. Self-supervised learning for ecg-based emotion recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3217–3221.
23. Franceschi, J.Y.; Dieuleveut, A.; Jaggi, M. Unsupervised scalable representation learning for multivariate time series. *arXiv* **2019**, arXiv:1901.10738.
24. Schneider, S.; Baevski, A.; Collobert, R.; Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv* **2019**, arXiv:1904.05862.
25. Dau, H.A.; Bagnall, A.; Kamgar, K.; Yeh, C.C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E. The UCR time series archive. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 1293–1305. [CrossRef]
26. Bagnall, A.; Dau, H.A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; Keogh, E. The UEA multivariate time series classification archive, 2018. *arXiv* **2018**, arXiv:1811.00075.
27. Fan, H.; Zhang, F.; Gao, Y. Self-Supervised Time Series Representation Learning by Inter-Intra Relational Reasoning. *arXiv* **2020**, arXiv:2011.13548.
28. Bengio, Y. *Learning Deep Architectures for AI*; Now Publishers Inc.: Norwell, MA, USA, 2009.
29. Oord, A.v.d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
30. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
31. Schmidt, P.; Reiss, A.; Duerichen, R.; Marberger, C.; Van Laerhoven, K. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In Proceedings of the 20th ACM International Conference on Multimodal Interaction, Boulder CO, USA, 16–18 October 2018; pp. 400–408.
32. Dosovitskiy, A.; Springenberg, J.T.; Riedmiller, M.; Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27* . [CrossRef] [PubMed]
33. Doersch, C.; Gupta, A.; Efros, A.A. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1422–1430.
34. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
35. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time series data augmentation for deep learning: A survey. *arXiv* **2020**, arXiv:2002.12478.

36. Um, T.T.; Pfister, F.M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 216–220.
37. Lei, Q.; Yi, J.; Vaculin, R.; Wu, L.; Dhillon, I.S. Similarity Preserving Representation Learning for Time Series Clustering. *arXiv* **2017**, arXiv:1702.03584.
38. Wu, L.; Yen, I.E.H.; Yi, J.; Xu, F.; Lei, Q.; Witbrock, M. Random warping series: A random features method for time-series embedding. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Lanzarote, Spain, 9–11 April 2018; pp. 793–802.
39. Lucas, B.; Shifaz, A.; Pelletier, C.; O'Neill, L.; Zaidi, N.; Goethals, B.; Petitjean, F.; Webb, G.I. Proximity forest: An effective and scalable distance-based classifier for time series. *Data Min. Knowl. Discov.* **2019**, *33*, 607–635. [CrossRef]
40. Lee, Y.K.; Kwon, O.W.; Shin, H.S.; Jo, J.; Lee, Y. Noise reduction of PPG signals using a particle filter for robust emotion recognition. In Proceedings of the 2011 IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 6–8 September 2011; pp. 202–205.
41. Liang, Y.; Elgendi, M.; Chen, Z.; Ward, R. An optimal filter for short photoplethysmogram signals. *Sci. Data* **2018**, *5*, 180076. [CrossRef]
42. Hanyu, S.; Xiaohui, C. Motion artifact detection and reduction in PPG signals based on statistics analysis. In Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 3114–3119.
43. Sadhukhan, D.; Pal, S.; Mitra, M. PPG Noise Reduction based on Adaptive Frequency Suppression using Discrete Fourier Transform for Portable Home Monitoring Applications. In Proceedings of the 2018 15th IEEE India Council International Conference (INDICON), Coimbatore, India, 16–18 December 2018; pp. 1–6.
44. Pollreisz, D.; TaheriNejad, N. Detection and removal of motion artifacts in PPG signals. *Mob. Netw. Appl.* **2019**, *27*, 728–738. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.