Journal of Intelligent Medicine and Healthcare
DOI: 10.32604/jimh.2022.036624
Article

ECG Heartbeat Classification Under Dataset Shift

Zhiqiang He*

Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing, 210044, China
*Corresponding Author: Zhiqiang He. Email: 20211249410@nuist.edu.cn
Received: 07 October 2022; Accepted: 15 November 2022

Abstract: Electrocardiogram (ECG) is widely used to detect arrhythmia. Atrial fibrillation, atrioventricular block, premature beats, etc. can all be diagnosed by ECG. When the distribution of training data and test data is inconsistent, the accuracy of the model will be affected. This phenomenon is called dataset shift. In the real-world heartbeat classification system, the heartbeat of the training set and test set often comes from patients of different ages and genders, so there are differences in the distribution of data sets. The main challenge in applying machine learning algorithms to clinical AI systems is dataset shift. Test-time adaptation (TTA) aims to adapt a pre-trained model from the source domain (SD) to the target domain (TD) without using any SD data or TD labels, thereby reducing model performance degradation due to domain differences. We propose a method based on multimodal image fusion and continual test-time adaptation (FCTA) for accurate and efficient heartbeat classification. First, the original ECG data is converted into a three-channel color image through a multimodal image fusion framework. The impact of class imbalance on network performance is overcome using a batch weight loss function, and then the pretrained source model is adapted to the TD using a continual test-time adaptation (CTA) method. Although our method is very simple, compared with other domain adaptation methods, it can significantly improve model performance on the test set and reduce the impact caused by the difference in domain distribution.

Keywords: ECG heartbeat classification; test-time adaptation; dataset shift

1 Introduction

The World Health Organization (WHO) survey on cardiovascular disease [1] shows that cardiovascular disease is still the world's first killer. In severe cases, coronary heart disease and stroke will cause arrhythmias, often sinus tachycardia or some occasional premature beats, and myocardial infarction. These symptoms can be monitored with portable single-lead ECG equipment. With the increase of various ECG-collecting devices and the collection of more and more ECG data, there is a growing interest in research using deep learning techniques to classify heartbeats. On the basis of the Association for the Advancement of Medical Instrumentation (AAMI) classification criteria, 14

subclasses can be integrated into 5 major classes: normal beat (N), supraventricular ectopic beat (S), ventricular ectopic beat (V), fusion beat (F), and unknown beat (Q).

Deep neural network models achieve outstanding performance in various deep learning applications when SD data and TD data have identical data distributions [2,3], but when there is a distributional difference between the training and testing distributions, i.e., when there is a dataset shift [4], model performance drops dramatically. Due to the dataset shift between SD data and TD data, if the pre-trained model is directly used for test data, the performance will be poor, so it is necessary to use the test time during inference to adapt from the unlabeled TD data learning to improve model performance on the test set. Most work in this field focuses on how to train a robust model during training [5–7], or retrain the model using source and TD data to adapt the model to changing TD data [8]. However, in the medical field, these methods are not feasible because many of the datasets used in training are private data of patients, which are usually unavailable during testing.

TENT [9] and TTT [8] are two very effective TTA methods. TENT updates model parameters through entropy minimization to adapt the model to distribution changes between SD and TD. The TTT uses auxiliary self-supervision tasks to train the source model and then fine-tune the model using the test data. These TTA methods are effective when the distribution of TD data is unchanged, but when the TD data comes from a changing domain, the model suffers from error accumulation in the process of adapting to the changing TD [10] and catastrophic forgetting [11,12], resulting in unstable model performance.

This paper aims to provide an efficient, stable, and practical ECG heartbeat classification method, which can be well used in the existing portable ECG equipment for monitoring of arrhythmia. First, the original ECG signal is directly converted into a 2D image through a multimodal fusion framework, thus eliminating complex data preprocessing operations. Moreover, the 2D image maintains the time dependence of the ECG signal without losing any information from the one-dimensional signal. Then, input the processed training data set into the convolutional neural network (CNN) for pre-training. Finally, during inference, use the CTA method to make the pre-trained model adapt to the test data continuously and reduce the performance degradation caused by data offset. The overall framework of our proposed FCTA method is shown in Fig. 1. We have made the following three contributions:

1. Apply the CTA method to the ECG dataset to reduce network model performance degradation due to dataset shift, solving the problem of error accumulation and catastrophic forgetting in the process of CTA.
2. The ECG signal can be used directly to achieve efficient, stable, and practical heartbeat classification without any data preprocessing, thus eliminating a lot of complex data preprocessing operations and greatly reducing the cost of data preprocessing.
3. Effectively solves the problem of poor classification effect of the model for small samples due to the imbalance of the ECG dataset.

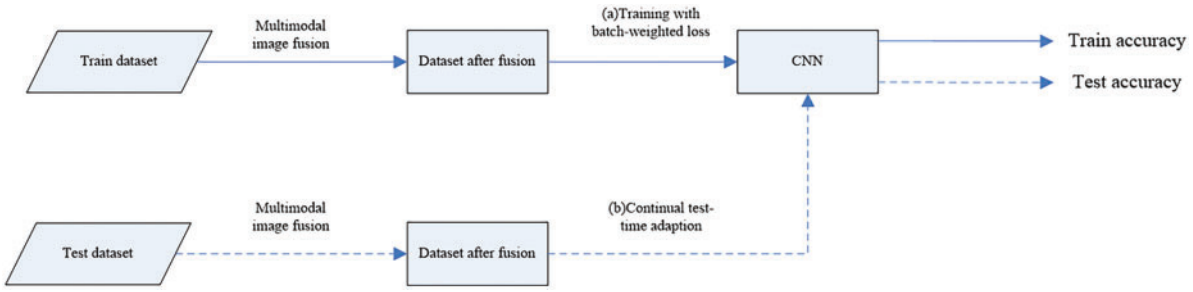

Figure 1: Method overview

2 Related Works

2.1 Continual Learning

Deep learning methods closely related to the problem of continual adaptation are continual learning [13] and lifelong learning [14], both of which can effectively alleviate the catastrophic forgetting of models. Continual learning methods are usually divided into regularization-based [15,16], methods based on replay [17,18], and parameter separation [19–21]. The regularization-based method adds some constraints when updating the parameters of the model, so that the model will not forget the knowledge previously learned when adapting to new tasks. Continual learning is mainly used to solve two problems. It can apply the experience of previous tasks so that the current task can be learned faster and better. When learning the current task, the task that has been learned before will not be forgotten, that is, to enhance the plasticity and stability of the model.

2.2 Test-Time Adaptation

General domain adaptation methods need to use SD data and TD data, while TTA does not need to use SD data but only uses source models pre-trained from SD data and unlabeled TD data, so test time adaptation is a very challenging task in the domain of domain adaptation. TENT uses the SD pretrained model for the TD and uses entropy minimization to update the trainable parameters in the Batchnorm layer [22] to fit the test data. The MEMO [23] is similar to TENT in that it updates the model parameters by minimizing the marginal entropy and increases the robustness of the model by using data enhancement, but updates the parameters of all layers of the model. SITA [24] only needs one test sample at a time during the adaptation process, avoiding the dependence on batch size in the TENT method. In SITA, only the mean and variance in the Batchnorm layer need to be modified during the adaptation process, without the need for reverse propagation to update other parameters. When the TD comes from the same distribution, these TTA methods can achieve considerable results. However, when the TD is constantly changing, the performance of the model is not stable. For example, in the automatic driving scene, the weather may change at any time. At this point, the model needs to deal with the continuously changing target domain.

2.3 Fusion Based Approaches

Deep learning network models for ECG heartbeat classification include 1D and 2D CNNs [25–27]. Existing work typically necessitates complex dataset preprocessing, such as data desiccation and feature extraction, before feeding the dataset into the model for training. Multimodality-based methods can fuse different modes and accurately classify tasks by integrating complementary information from modalities [28]. Reference [29,30] proposed a Multi-scale Fusion CNN (MS-CNN)

and a Deep Multi-scale Fusion CNN (DMSFNet) for arrhythmia detection, respectively, and in [31], a multilevel feature fusion framework based on CNN and an attention module is proposed. It extracts features from different layers of CNN to perform classification and improves the recognition ability of the ECG classification model by combining CNN and the attention module.

3 Methodology

3.1 Problem Statement

Given a source model f_θ pre-trained on the SD $\left(X^S, Y^S\right)$, the model has a poor classification effect on data from different distributions in the SD. Our goal is to improve the performance of the source model on the changing TD $\left(X^T, Y^T\right)$ at inference time. During CTA, the source data of the SD is not available. Only the pre-trained source model and unlabeled TD data X^T are available.

In Table 1, we list the differences between the CTA and the existing domain adaptation methods. Compared with the previous settings, the CTA focuses more on the continual adaptation of the changing TD scenarios.

Table 1: Differences in settings between TTA and other adaptation methods

Setting	Source data	Target data	Train loss	Test loss
Fine-tuning	×	x^t,y^t (stationary)	$L\left(x^t,y^t\right)$	×
Domain adaption	x^s,y^s	x^t (stationary)	$L\left(x^s,y^s\right)+L\left(x^s,x^t\right)$	×
TENT	×	x^t (stationary)	×	$L\left(x^t\right)$
FCTA (ours)	×	x^t (changing)	×	$L\left(x^t\right)$

3.2 Multimodal Image Fusion

Gramian Angular Field (GAF), recursive graph (RP), and Markov Transition Field (MTF) images were created from the one-dimensional ECG data. Then, the three gray images are combined into three-channel color images (GAF-RP-MTF), which are formed from the original ECG data by different statistical methods and maintain the signal-dependent time without losing any information of the one-dimensional signal. Therefore, the obtained three-channel color image contains more information, and the three-channel image can be easily used with off-the-shelf CNNs such as AlexNet [32] and ResNet.

3.3 Batch Weight Loss Function for Imbalanced ECG Data

Table 2 shows the category distribution of the PhysioNet MIT-BIH Arrhythmia Database (MIT-BIH) [33]. It is obvious that there is a class imbalance problem. The five categories N, S, V, F, and Q in the dataset account for 82.77%, 2.54%, 6.62%, 0.73%, and 7.34%, respectively. For the sake of solving the class imbalance problem in the dataset, Z. Ahmad et al. [28] used the SMOTE algorithm [34] to upsample the samples of four classes except class N, but there are some differences between the samples sampled by the SMOTE algorithm and the original samples. In the training process, the method we propose only uses the raw ECG data without adding additional data.

Table 2: The number of heartbeats of different categories in the MIT-BIH database

Dataset	Classes	Number of beats	Percentage (%)
MIT-BIH	N	90587	82.77
	S	2781	2.54
	V	7245	6.62
	F	802	0.73
	Q	8038	7.34

To overcome the impact of dataset imbalance on model performance, during training, we use a batch weight loss function [35]. Then, define the label set of N heartbeats in the ith batch as $Batch\_labels_i = \left\{y^s_{i,1}, y^s_{i,2}, \ldots, y^s_{i,N}\right\}$, where $y^s_{i,j} \in \{N,S,V,F,Q\}$, then, defined in the ith batch loss weight of each category is $W_{i,class_c}$, where $class_c \in \{N,S,V,F,Q\}$.

$$W_{i,class_c} = 1 - \frac{\sum_{j=1}^{N} I_{y_{i,j}=class_c}}{N} \tag{1}$$

where N is the batch size and I is the indicator function, When $y_{i,j} = class_c$, the value of the indicating function is 1, otherwise it is 0. After determining the loss weight of each class, the weighted loss function of the ith batch is shown in Eq. (2).

$$L_i^{train} = -\sum_{j=1}^{N} W_{i,y^s_{i,j}} y^s_{i,j} \log p^s_{i,j} \tag{2}$$

where $y^s_{i,j}$ is the label of the jth training sample in the ith batch, and $p^s_{i,j}$ is the prediction probability of the model for the jth training sample in the ith batch.

3.4 Continual Test-Time Adaption

The CTA method [36] consists of three parts, namely, exponential moving average pseudo-labels, data augmentation average pseudo-labels, and random recovery weight. The first two parts are to alleviate the accumulation of errors in the model due to the use of pseudo-labels. Exponential moving average pseudo-labels and data augmentation average pseudo-labels can improve the quality of pseudo-labels, thereby improving model performance. Randomly restoring source model weights is to recover any weights of the model as initial parameters in the process of continuously adapting to new TD data so as to mitigate catastrophic forgetting of the model.

3.4.1 Exponential Moving Average Pseudo-Labels

Given TD data x^t and a source model f_θ, the goal of TTA is to minimize the entropy loss between the model's predictions $y = f_\theta(x^t)$ for the given target data and the pseudo-labels. In TENT, the predictions of the model itself are directly used as pseudo-labels. Reference [37] proved that pseudo-labels averaged by weights work better than directly using model predictions as pseudo-labels. Therefore, We update the weights of the teacher model (TM) by using the weighted exponential moving average of the student model (SM), and then use the predictions of the updated TM as pseudo labels. In the method of continually testing time domain adaptation, there are pretrained source model f_θ, TM f_{θ_t} and SM f_{θ_s}. At time step t = 0, initialize TM and SM with source model parameters, at time step t,

the predicted results of the TM are used as pseudo-labels. Then use Eq. (3) to find the cross-entropy loss predicted by the TM and the SM. Back-propagate to update the parameters of the SM θ_s. After updating the SM weights ($\theta_s \rightarrow \theta_{s+1}$), use the exponential moving average of the SM parameters to update the weights of the TM ($\theta_t \rightarrow \theta_{t+1}$), the exponential moving average update rule is as Eq. (4).

$$L_{\theta_s}\left(x^t\right) = -\sum_{C=1}^{M} y_{tc} \log y_{sc} \tag{3}$$

where y_{tc} and y_{sc} represents the probability that a sample belongs to class c in the prediction of TM and SM, respectively, and M is the number of classes in the dataset.

$$\theta_{t+1}^{'} = (1-\alpha)\,\theta_t^{'} + \alpha\theta_{t+1} \tag{4}$$

where θ_{t+1} and $\theta_{t+1}^{'}$ represent the parameters of the SM and the TM when the time step is t + 1, respectively, α is the hyperparameter smoothing factor of the exponential moving average, and the class with the largest probability in the output probability y_{tc} of the TM is the class predicted by the model for the target data x^t.

The weighted average pseudo-label updates the TM through the exponential moving average SM and obtains more accurate pseudo-labels through the prediction of the TM better performance.

3.4.2 Data Augmentation Average Pseudo-Labels

Data augmentation is widely used to improve the generalization and robustness of the model during training, and different data augmentation needs to be set for different datasets. In reference [38,39], it was demonstrated that data augmentation at test time can also improve the robustness of the model, but it is not possible to set a fixed data augmentation strategy for changing TD. In this paper, we quantify domain differences by calculating model confidence and choose whether to use data augmentation according to the size of domain differences. First, use the source model f_θ to find the entropy value for the current input x^t as the model confidence. The confidence is calculated as Eq. (5). When the model confidence *confidence* (f_θ) is greater than the confidence threshold β_{th}, directly use the prediction y_{tc} of the TM is used as a pseudo-label without any data augmentation, and when the confidence is less than the confidence threshold β_{th}, an additional N random data augmentation strategies are used. Attempts to approximate the domain difference by predicting confidence, with the assumption that lower confidences represent larger domain differences and higher confidences represent smaller domain differences. The data augmentation strategy used in this paper is weak augmentation and strong augmentation, weak augmentation is a jitter and scale strategy, which adds random changes to the signal and amplifies its amplitude. Strong augmentation is a permutation and jitter strategy, which first divides the signal into a random number of segments and disrupts the order, and then adds random jitter to the replacement signal [40].

$$confidence\,(f_\theta) = -\sum_{C=1}^{M} f_\theta\left(x^t\right) \log f_\theta\left(x^t\right) \tag{5}$$

$$\overline{y_{tc}} = \frac{1}{N}\sum_{i=0}^{N-1} f_{\theta_t}\left(augment\left(x^t\right)\right) \tag{6}$$

$$y_{tc}^{'} = \begin{cases} y_{tc}, & confidence\,(f_\theta) \geq \beta_{th} \\ \overline{y_{tc}}, & otherwise \end{cases} \tag{7}$$

$$L_{\theta_s}\left(x^t\right) = -\sum_{C=1}^{M} y'_{tc} \log y_{sc} \tag{8}$$

where $\overline{y_{tc}}$ is the prediction of the TM after data augmentation, y_{tc} is the prediction of the TM without data augmentation. After using the augmented average pseudo-label, the cross-entropy loss is changed from Eq. (3) update to Eq. (8).

3.4.3 Random Recovery Weight

In the process of continual adaptation, the model will continue to learn new TD knowledge and gradually forget the knowledge from the SD. In practical application, we hope that the model can not only adapt to new tasks quickly but also not forget the knowledge learned before. We use a simple and efficient random recovery method to mitigate the catastrophic forgetting of the model. This method preserves part of the knowledge from the source model by randomly recovering weights from it. The weight recovery rules are shown in Eq. (9).

$$W_{t+1} = (1 - N) * W_0 + N * W_{t+1} \tag{9}$$

$$N \sim Bernoulli\,(p) \tag{10}$$

where $*$ represents the element-by-element multiplication between the two matrices, p is the probability of random recovery, N is the mask tensor conforming to the Bernoulli distribution, W_0 is the convolutional layer convolution kernel weight of the source model, W_{t+1} is the convolutional layer convolutional layer convolution kernel weight at time step t + 1. By randomly restoring a small number of tensors in the trainable weights to the initial weights before adaptation, random recovery avoids that the model completely forgets the knowledge learned before adapting to new tasks, thus mitigating the catastrophic forgetting of the model.

4 Experiments

4.1 Database and Experimental Setup

In this experiment, the training set and test set are MIT-BIH and PhysioNet MIT-BIH ST Change Database (MIT-BIH-ST), respectively. According to AAMI classification criteria, the database is divided into five categories. Since there are no F and Q labels in the test set, we can finally only classify N, S, and V into three categories. In this experiment, both training set and test set are resampled to 360 Hz. Our method can be used with any existing deep neural network without modifying the existing network. In this experiment, we use the ResNet-18 and WideResNet-28 [41] network structures. In the process of training a source model, the batch size is 128, the learning rate is 0.01, and the number of iterations in the learning process is 30.

4.2 Experiment Results

The indicators used for classification in this experiment are accuracy (acc), precision (pre) and recall (rec). The calculation methods for accuracy, precision and recall are as follows:

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$pre = \frac{TP}{TP + FP} \tag{12}$$

$$rec = \frac{TP}{TP + FN} \tag{13}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

4.2.1 Experiments on ResNet-18

The source model achieves 97.78% acc, 96.28% pre, and 96.82% rec on the training set. Although the source model performs well on the training set, if the source model is directly used for the test set, there is a dataset shift between the training set and the test set, and the model performance during testing is very poor. After using FCTA, the performance improvement effect is significant. In order to compare the FCTA method with TENT, BN [42], We follow the TTA settings in TENT. In the TTA process, we chose SGD as the optimization algorithm. The learning rate is 0.001, the batch-size is 32, the number of random data augmentations N is 32, the confidence threshold β_{th} is 0.9, and the random recovery probability p is 0.05. Table 3 shows a comparison of the performance of the FCTA method with several other adaptation methods.

Table 3: Experimental results on MIT-BIH-ST database using ResNet-18

Method	acc%	pre%	rec%
Source	59.39	58.56	60.26
BN [42]	71.23	68.21	69.31
TENT [9]	73.21	70.32	70.12
FCTA (ours)	80.14	79.02	78.27

To verify the unique advantages of the FCTA method in continual learning and mitigating catastrophic forgetting, we reuse the test-time-adapted model on the test training set to verify that the model can also retain the knowledge learned in the training set. Table 4 shows the performance of ResNet-18 on the MIT-BIH database before and after the TTA. Obviously, after using TENT and BN adaptation, the model suffers from catastrophic forgetting, while the FCTA method preserves the source model most of the knowledge.

Table 4: Experimental results on MIT-BIH database using ResNet-18

Method	acc before adaption	acc after adaption
BN [42]	97.78	73.21
TENT [9]	97.78	62.48
FCTA (ours)	97.78	95.46

4.2.2 Experiments on WideResNet-28

The FCTA method can be applied to any existing model without the need to adjust the existing model. For the sake of verifying the robustness of the duration adaptation method to the model, we apply it to the WideResNet-28. The source model achieves 99.12% acc, 98.37% pre, and 97.45% rec on the training set. In the FCTA setting, the parameters are the same as in the experiments using ResNet-18. Table 5 shows the experimental results.

Table 5: Experimental results on MIT-BIH-ST database using WideResNet-28

Method	acc%	pre%	rec%
Source	68.34	67.21	67.56
BN [42]	73.51	73.56	72.31
TENT [9]	75.86	74.97	74.64
FCTA (ours)	82.34	81.53	80.91

Table 6 shows the performance of the WideResNet-28 model on the MIT-BIH database before and after TTA, which verifies that our method can effectively solve the problem of error accumulation and catastrophic forgetting in the process of FCTA.

Table 6: Experimental results on MIT-BIH database using WideResNet-28

Method	acc before adaption	acc after adaption
BN [42]	98.32	76.83
TENT [9]	98.32	67.97
FCTA (ours)	98.32	96.54

5 Conclusion

We presented FCTA, a method based on multimodal image fusion and continual test-time adaptation for ECG classification tasks, focusing on how to address the model's error accumulation and catastrophic forgetting during CTA. First, the original ECG dataset is converted into a two-dimensional image through a multimodal image fusion framework. For the sake of alleviating the class imbalance of the ECG database, a batch weight average loss function is used when pretraining the model. Second, the pseudo-label quality is improved by the exponential moving average and data augmentation average, thereby reducing error accumulation. Finally, for the purpose of mitigating the catastrophic forgetting of the model, after each adaptation, a part of the parameters of the model are randomly restored to the parameters of the source model, and the knowledge learned by the source model is retained. The FCTA approach can be applied to any ready-made pre-training model, and compared to using the pre-trained model directly at test time, after using FCTA, the performance of the model on the TD is significantly improved, proving the effectiveness of our proposed method.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] WHO and others, "Cardiovascular diseases," 2020. https://www.who.int/zh/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019.

[2] K. M. He, X. Y. Zhang, S. Q. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.

[3] J. D. Deng, S. Wei, R. Li, L. J. Li, and F. F. Li, "A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, Florida, USA, pp. 248–255, 2009.
[4] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, 2012.
[5] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer *et al.,* "Augmix: A simple data processing method to improve robustness and uncertainty," arXiv preprint arXiv:1912.02781, 2019.
[6] F. C. Qiao, L. Zhao, and P. Xi, "Learning to learn single domain generalization," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 12556–1265, 2020.
[7] S. Sagawa, P. W. Koh, T. B. Hashimoto and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," arXiv preprint arXiv:1911.08731, 2019.
[8] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros *et al.,* "Test-time training with self-supervision for generalization under distribution shifts," in *Int. Conf. on Machine Learning*, Vienna, Austria, pp. 9229–9248, 2020.
[9] D. Wang, E. Shelhamer, S. Liu, B. Olshausen and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," arXiv preprint arXiv:2006.10726, 2020.
[10] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger, "On calibration of modern neural networks," in *Int. Conf. on Machine Learning*, Sydney, Australia, pp. 1321–1330, 2017.
[11] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell and M. Rohrbach, "Adversarial continual learning," in *European Conf. on Computer Vision*, Glasgow, UK, pp. 386–402, 2020.
[12] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.
[13] M. De Lange, R. Alijundi, M. Masana, S. Parisot, X. Jia *et al.,* "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
[14] M. Laal and P. Salamati, "Lifelong learning; why do we need it?" *Procedia-Social and Behavioral Sciences*, vol. 31, pp. 399–403, 2012.
[15] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci *et al.,* "Class-incremental learning via deep model consolidation," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, Snowmass Village, Colorado, US, pp. 1131–1140, 2020.
[16] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins *et al.,* "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
[17] M. De Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, Waikoloa, HI, US, pp. 8250–8259, 2021.
[18] H. Shin, J. K. Lee, J. Kim and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, Cambridge, MA, UK, pp. 1253–1264, 2017.
[19] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7765–7773, 2018.
[20] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *European Conf. on Computer Vision*, Munich, Germany, pp. 67–82, 2018.
[21] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick *et al.,* "Progressive neural networks," arXiv preprint arXiv:1606.04671, 2016.
[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, Lille, France, pp. 448–456, 2015.
[23] M. Zhang, S. Levine and C. Finn, "Memo: Test time robustness via adaptation and augmentation," arXiv preprint arXiv:2110.09506, 2021.
[24] A. Khurana, S. Paul, P. Rai, S. Biswas and G. Aggarwal, "Sita: Single image test-time adaptation," arXiv preprint arXiv:2112.02355, 2021.

[25] L. Sun and J. Wu, "A scalable and transferable federated learning system for classifying healthcare sensor data," *IEEE Journal of Biomedical and Health Informatics*, 2022. https://doi.org/10.1109/JBHI.2022.3171402.
[26] L. Sun, Z. Zhong, Z. Qu and N. Xiong, "PerAE: An effective personalized AutoEncoder for ECG-based biometric in augmented reality system," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 6, pp. 2435–2446, 2022.
[27] Y. X. Zhang and Y. Zhou, "Resource allocation strategy based on tripartite graph in vehicular social networks," *IEEE Transactions on Network Science and Engineerin*, 2022. https://doi.org/10.1109/TNSE.2022.3153511.
[28] Z. Ahmad, A. Tabassum, L. Guan and N. M. Khan, "ECG heartbeat classification using multimodal fusion," *IEEE Access*, vol. 9, pp. 100615–100626, 2021.
[29] X. Fan, Q. Yao, Y. Cai, F. Miao, F. Sun *et al.,* "Multiscaled fusion of deep convolutional neural networks for screening atrial fibrillation from single lead short ecg recordings," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1744–1753, 2018.
[30] R. Wang, J. Fan, and Y. Li, "Deep multi-scale fusion neural network for multi-class arrhythmia detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2461–2472, 2020.
[31] R. Wang, Q. Yao, X. Fan, and Y. Li, "Multi-class arrhythmia detection based on neural network with multi-stage features fusion," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, Bari, Italy, pp. 4082–4087, 2019.
[32] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[33] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote:Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[35] A. Sellami and H. Hwang, "A robust deep convolutional neural network with batch-weighted loss for heartbeat classification," *Expert Systems with Applications*, vol. 122, pp. 75–84, 2019.
[36] Q. Wang, O. Fink, L. Van Gool and D. Dai, "Continual test-time domain adaptation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, US, pp. 7201–7211, 2022.
[37] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, Cambridge, MA, UK, pp. 1195–1204, 2017.
[38] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM Journal on Control and Optimization*, vol. 30, no. 4, pp. 838–855, 1992.
[39] G. Cohen and R. Giryes, "Katana: Simple post-training robustness using test time augmentations," arXiv preprint arXiv:2109.08191, 2021.
[40] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh *et al.,* "Time-series representation learning via temporal and contextual contrasting," arXiv preprint arXiv:2106.14112, 2021.
[41] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint arXiv:1605.07146, 2016.
[42] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel *et al.,* "Improving robustness against common corruptions by covariate shift adaptation," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 11539–11551, 2020.