SN Computer Science (2022) 3:70
https://doi.org/10.1007/s42979-021-00924-x

ORIGINAL RESEARCH

Data Augmentation for 12-Lead ECG Beat Classification

Edmund Do[1] · Jack Boynton[2] · Byung Suk Lee[1] · Daniel Lustgarten[3]

Received: 23 January 2021 / Accepted: 4 October 2021

Abstract
This paper reports the performance study of classifying 12-lead ECG beat segments in the face of severe imbalance in the class sizes, which is typical of ECG data. The efficacy of data augmentation for class size balancing to improve the classification accuracy is well known. In the ECG domain, however, it has been overlooked or handled inadequately. We propose an amplitude-alteration approach to augment randomly selected ECG heartbeats separately as needed in individual ECG classes while not disrupting the timeline of the ECG signals. In addition, augmentations of training dataset and test dataset receive separate attentions, and four cases of data augmentation are considered depending on whether each dataset is augmented or not. The effects of the augmentation scheme was evaluated using ResNet, the deep learning technique reputed for its remarkable accuracy through unique skipping connections between layers; specifically, a time series version of ResNet was used. The results confirmed the key benefit of class-balancing the training dataset through the proposed data augmentation scheme and, additionally, showed some extra benefit of augmenting the test dataset through "test-time augmentation." Further, we adopted the class activation map (CAM) to identify heat map signatures that would explain ECG beat classes. The results demonstrated that the CAM could be an effective visual aid to classifying ECG beats, especially with the proposed data augmentation scheme in place. In this paper the augmentation scheme, the associated experiments, and their results are discussed concretely.

Keywords 12-Lead ECG · Classification · Class balancing · Data augmentation · Time series ResNet · Class activation map · Performance study

Introduction

Computer-aided electrocardiogram (ECG) classification is a diagnostic tool for identifying the abnormality types (or normality) of individual heartbeats or a set of heartbeats (da Luz et al. [1]) using machine learning algorithms. The inherent problem with ECG datasets is the imbalance in class sizes, wherein some classes have plenty of beats while others have few. This is problematic for machine learning algorithms since trained classifiers typically have high error rates for sparse classes (Chawla et al. [2]). One natural resolution is data augmentation for sparse classes where ECG beats are altered without affecting the beat type to balance class sizes. Models trained with augmented data typically result in improved accuracy and reduced generalization error.

It is increasingly important that we train classifiers properly since ECGs can be used for various tasks in areas such as health monitoring (Hwang et al. [3]), (Manogaran et al. [4]),(Trenta et al. [5]), medical diagnoses (Kwon et al. [6]), (Han and Shi [7]), ECG classification and annotation (He et al. [8]), (Tung et al.[9]), and biometric authentication (Labati et al. [10]).

Recently, deep learning has emerged as an effective technique to ECG classification (Aurore et al. [11]). One of the best-performing deep learning architectures for time series is

✉ Byung Suk Lee
bslee@uvm.edu

Edmund Do
edo@uvm.edu

Jack Boynton
jwboynto@uvm.edu

Daniel Lustgarten
dlustgar@uvm.edu

[1] Department of Computer Science, University of Vermont, Burlington, Vermont, USA

[2] Department of Electrical and Biomedical Engineering, University of Vermont, Burlington, Vermont, USA

[3] Department of Medicine, University of Vermont, Burlington, Vermont, USA

Published online: 19 November 2021

ResNet (He et al. [12]). While ResNet was originally introduced for image classification, Wang et al. [13] explored using it for time series classification, and outstanding performance has been demonstrated by Ismail Fawaz et al. [14].

The primary objective of our work is to handle the problem of class imbalance in an ECG dataset through our novel ECG data augmentation scheme. We evaluate the effect of data augmentation on the classification accuracy of the time series ResNet for a "general" purpose; that is, to classify *full 12-lead* ECG beats into all *individual symptom* classes—not "superclasses"— available from the PhysioBank ECG datasets annotated (PhysioNet [15]). [This 12-lead ECG classification is the topic of a recently announced PhysioNet Computing Challenge (Alday et al. [16])].

To this end, the key ideas of our ECG data augmentation method are (1) altering only the amplitudes (and keeping the timeline intact), (2) altering individual ECG beats separately and independently, and (3) altering each beat across the 12 leads in the same way. Currently existing ECG data augmentation methods (Pan et al. [17]), (Cui et al. [18]), (Um et al. [19]), (Cao et al. [20]), (Le Guennec et al. [21]), (Jun et al. [22]), (He et al. [8]), (Yao et al. [23]), (Acharya et al. [24]) are inadequate or limited for use for our purpose (i.e., classifying 12-lead ECG beats into 9 classes shown in Table 3), as discussed in "ECG Data Augmentation".

A thorough performance study was conducted comparing four cases of ECG data augmentation depending on whether each of the training dataset and the test dataset is augmented or not. The results confirmed that augmenting the *training* dataset toward balanced classes using the proposed augmentation scheme is the key to achieving high classification accuracy; in addition, augmenting the test dataset (called "test-time augmentation"—more in "Test-Time Augmentation") using the proposed augmentation scheme was helpful when the accuracy metrics are sensitive to the class balancing (e.g., macro F1), especially when accompanying the augmentation of the training dataset.

Further, the class activation map (CAM) was applied to the ECG time series data. CAM was originally introduced for images by Zhou et al. [25] and applied to time series by Wang et al. [13], and then was used for ECGs in a few related works by Goodfellow et al. [26], Oh et al. [27], and Wang et al. [28]. CAM successfully visualized an ECG heartbeat using a heatmap pattern, to enable identifying the heatmap signatures associated with individual diagnostic classes of beats. We examined the CAMs of classified heartbeats to visually identify the features of ECG beat signals that explain the output classes, both with and without the data augmentation. The results demonstrated the efficacy of CAM in verifying the classification accuracy visually, especially when the data augmentation is present.

To the best of our knowledge, this paper is the first that provides an effective ECG data augmentation scheme with a thorough study focused on the effects of data augmentation on the ECG heartbeat classification accuracy. The performance study results demonstrate the efficacy of the proposed ECG data augmentation scheme.

Table 1 lists the acronyms and abbreviations of key terms used in this paper. The Python source codes of ECG signal filtering and segmentation, data augmentation, ResNet modeling, and class activation mapping are available at the GitHub supplement (Boynton [29]).

The remainder of the paper is organized as follows. "Related Work" discusses relate work. "ECG Data" describes the ECG data used in this work. "Methods" discusses the methods used in the performance study. "Evaluations" presents the experiments and the results. "Conclusion" summarizes the paper and suggests further work.

Related Work

Let us discuss related work in two directions: ECG data augmentation and ECG data classification using machine learning.

ECG Data Augmentation

Despite the importance of class balancing through data augmentation to the quality of classification output, data augmentation in ECG has been quite limited, as was also pointed out by [17]. Pan et al. [17] applied time series alteration methods to augment ECG data and compared the

Table 1 Acronyms and abbreviations

Category	Acronym/ abbreviation	Description
Deep learning	CAM	Class activation map
	CNN	Convolutional neural network
	LSTM	Long short-term memory
	ReLU	Rectified linear unit
	ResNet	Residual network
	RNN	Recurrent neural network
	TTA	Test-time augmentation
ECG beat classes	APC	Atrial premature complex
	MI	Myocardinal infarction
	NESC	Nodal escape beat
	PVC	Premature ventricular complex
	RBBB	Right bundle branch block
	SVESC	Supraventricular escape beat
	SVPB	Supraventricular premature beat

efficacy of four such methods—window slicing (Cui et al. [18]), permutation (Um et al. [19]), concatenation and resampling (Cao et al. [20]), and window warping (Le Guennec et al. [21]). Pan et al. [17] evaluated these methods using LSTM RNN and observed good results, but the data augmentation and classification was done on *the entire ECG time series*, and thus their methods are not applicable to our work which needs to augment data at the level of individual ECG beats.

There are a few known works on ECG data augmentation and classification at *the beat level*. Jun et al. [22] rendered 1D ECG signals to 2D images and used image cropping and masking for use with CNN. This method is not applicable in our work which alters ECG as a 1D signal. He et al. [8] divided each class into five subsets and duplicated classes to match the number of samples in the most dominant class. The classification accuracy measured on nine classes in the 12-lead China Physiological Signal Challenge (CPSC) dataset was 80.6%. While this method may be able to balance the dataset sizes across classes, the trained classifier may not be robust to classes containing data duplicated from the ECG of a small number of patients.

Yao et al. [23] compressed or stretched the ECG signal along the timeline and removed parts of the segments (or beats). The classification accuracy measured on nine classes in the 12-lead CPSC dataset was 81.2%. This method is problematic because the ECG signal is a time series, and, therefore, its diagnostic classification is sensitive to alteration along the timeline. Note that our work uses an *amplitude* scaling method, which limits the signal alterations only to the amplitude, not the timeline.

Acharya et al. [24] altered samples by amplitude scaling, too. They varied the standard deviation and the mean of Z-scores from the original normalized ECG segments (of length 260 samples). The specifics of the implementing the scaling, however, is not stated, while it appears randomly selected scale factor was applied to each segment. The classification accuracy measured on five classes in one lead ECG was 94.03% with noise removal. While these results are quite good, the setup is imited compared with than our work where nine classes in 12 lead ECG are evaluated.

ECG Classification Using Machine Learning

Machine learning algorithms that can be trained to classify ECG heartbeat data include support vector machines (Rajesh and Dhuli [30]), random forests (Rahman et al. [31]), and hidden Markov models (Liang et al. [32]). When applied to 12-lead ECG data, most of them achieve classification accuracy above 90% (Lyon et al. [33]).

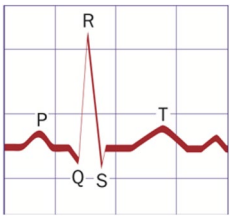
Since then, deep learning emerged with performance on par or better than the traditional machine learning algorithms. There have been several deep learning architectures used to classify ECG beat types, such as CNNs (Li et al. [34], (Yao et al. [23]), RNNs (Yildirim [35], Saadatnejad et al [36], Hong et al. [37]), and autoencoders (Hong et al. [37]). In addition, hybrid architectures combining CNN and RNN are used to combine CNN's ability to recognize features over space and RNN's ability to recognize patterns over time (Liu et al. [38], Lynn et al. [39]).

Only a few ECG classification studies (Han and Shi [7], Zhou et al. [40], Tung et al. [9]) used ResNet; and they restricted the study in either the number of ECG leads or the number of classes or both. Tung et al. [7] used ResNet to classify 12-lead ECG, but limited the classes to those of myocardial infarction (MI) based on the five types of MI location; their main interest was to compare the classification accuracy between intra- and inter-patient cases. He et al. [8] used ResNet with a bidirectional LSTM to classify 12-lead ECG beats in the CPSC dataset. Tung et al. [9] used ResNet embedded with attention blocks to classify two-lead ECG beats from the MIT-BIH (Moody and Mark [41]) arrhythmia database of two-lead (i.e., V1 and MLII) ECGs and utilized five beat types according to the AAMI standard (American National Standard [42]) (i.e., five superclasses mapped from 17 arrhythmia classes); their main interest was to combine ResNet with another architecture (i.e. attention blocks) for accuracy improvement. While all these work achieved outstanding classification accuracy (see Table 2), the extent of their studies was limited, tailored to their specific purposes. Compared with these studies, our work is more comprehensive and classifies nine classes (including Unknown beats) on 12-lead ECG.

Table 2 ECG classification accuracy of related works using ResNet

Study	Database	# Samples	# Classes	Accuracy
Han and Shi [7]	PTB	24157	5	Inter-patient: 95.49% Intra-patient: 99.92%
Tung et al. [9]	MIT-BIH	90007	5	97.15%
He et al. [8]	CPSC	9831	10	80.6%

Fig. 1 ECG beat segment



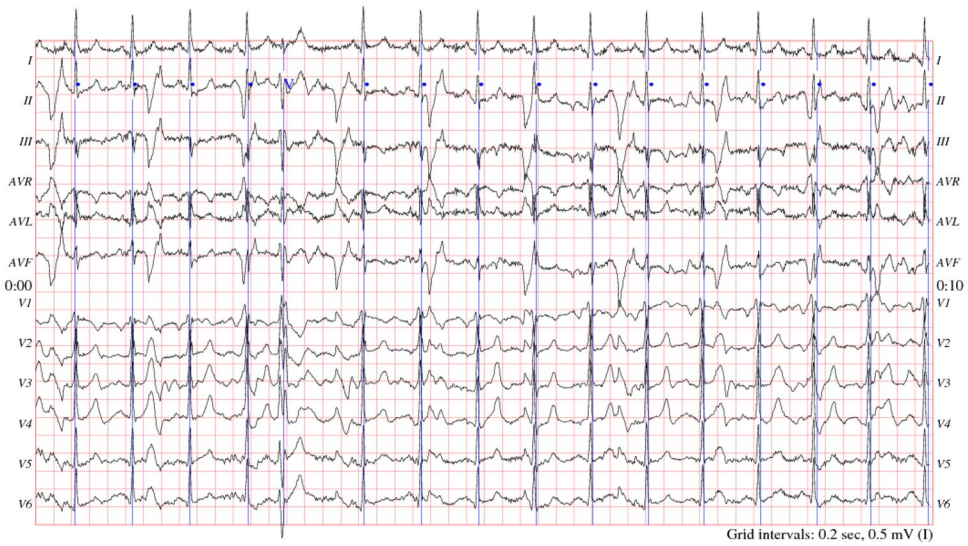


Fig. 2 12 lead ECG records (source: PhysioNet [43])

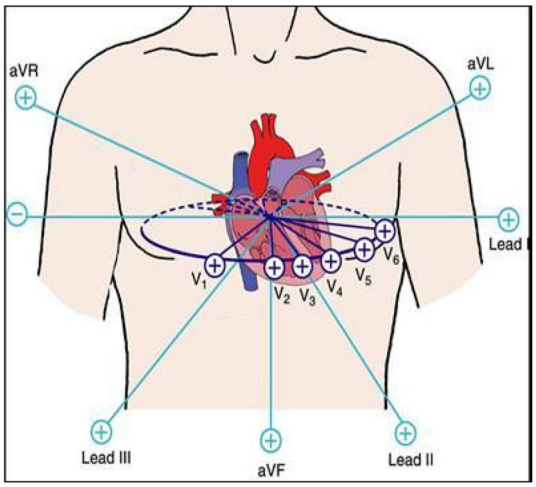


Fig. 3 12 lead ECG placement (source: Randazzo [44])

ECG Data

An ECG is an electrical signal manifesting the heartbeat over time. Figure 1 illustrates the composition of a single channel ECG segment. A normal ECG beat consists of a *P* wave, a QRS complex, an S wave, and a *T* wave in sequence. The *P* wave represents depolarization of the atria (i.e., top chambers of a heart); the QRS complex represents depolarization of the ventricles (i.e., bottom chambers of a heart); and the *T* wave represents repolarization of the ventricles. So, anomalous shapes of the components indicate certain clinical heart problems.

Figure 2 shows standard ECG signals (PhysioNet [43]) measured from 12 leads placed as illustrated in Fig. 3 (Randazzo [44]). There are six chest leads (V1 through V6) and six limb leads (I, II, III, aVR, aVL, and aVF). The limb leads provide a view of the heart from the front; among them, I, II, and III are bipolar and measure the electrical differences between the combination of three limbs, namely, the right arm, left arm, and left foot (here, bipolar means having a positive and negative pole in the electrical current measurement); and aVR, aVL, and aVF are unipolar and measure the electrical difference between the right and left arms and the left foot utilizing a central negative lead (Lieberman [45]).

The St. Petersburg INCART 12-lead ECG datasets (PhysioNet [43]) from PhysioNet (Goldberger et al. [46]) is used in this study. (There are five 12-lead ECG classification datasets published in PhysioNet Challenge 2020, but among them St. Petersburg INCART data set is the only

Table 3 INCART beat class distribution

Normal (%)	PVC (%)	RBBB (%)	APC (%)	Fusion (%)	NESC (%)	SVESC (%)	SVPB (%)	Unknown (%)
85.51	11.38	1.80	1.11	0.125	0.052	0.018	0.009	0.003

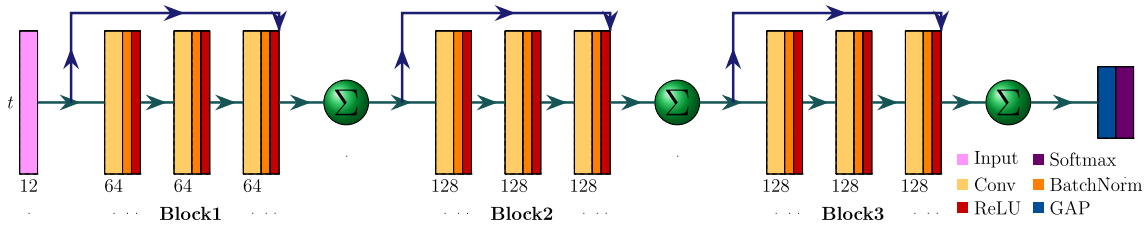


Fig. 4 Time series ResNet with a 12-channel, variable length input. The residual connections (blue arrows) allow the input to skip over the layers within a block

one that has been annotated with class labels for *each ECG segment*.) The 75 Holter records in this database come from 32 patients with various diagnosed heart complications including ischemia, coronary artery disease, conduction abnormalities, and arrhythmia. Each record is a time series sampled at 257 Hz and is approximately 30 min long. The heartbeats in each record are annotated with diagnostic classes in the PhysioNet annotations (PhysioNet [15]). There are nine diagnostic classes appearing in the ECG dataset, and their representations are summarized in Table 3. Notably, the class sizes are severely imbalanced, with Normal class beats predominant and some classes (e.g., Unknown, SVPB, SVESC, NESC) constituting less than 0.1% of the entire population of heartbeats.

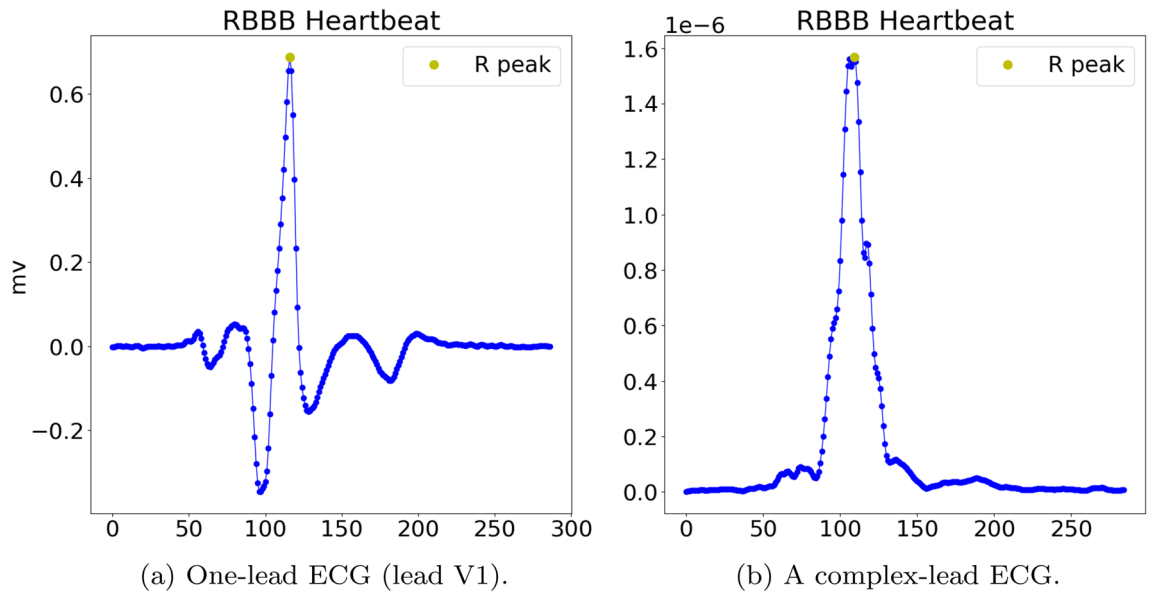
In our work, 12-lead ECG records are represented as a sequence of segments, one segment per heartbeat. A segment can be represented as a matrix $S \in \mathbb{R}^{c \times t}$ where $c\,(=12)$ is the number of channels and t is the length of the segment.

The individual leads' ECG records were normalized to have mean 0.0 and standard deviation 1.0. This normalization is commonly done in deep learning to, for instance, to achieve faster model convergence (Shimodaira [47]).

Methods

Time Series Residual Network

Convolutional layers are known to detect location-invariant features. In time series with similar features across samples, such as ECG, these location-invariant features become important to distinguish between diagnoses (i.e., classes). In conjunction with residual connections, the neurons in deeper



(a) One-lead ECG (lead V1). (b) A complex-lead ECG.

Fig. 5 An example of an extracted ECG beat segment. (See the supplement (Boynton [29]) for the full 12-lead beat segments)

layers can learn more abstract features to help distinguish samples that appear similar.

Network depth is crucial to learning higher-level representations of the data; however, deep neural networks often have difficulty to train due to the vanishing gradient and the resulting degradation of classification accuracy (He et al. [12]). ResNet overcomes this problem without increasing the number of parameters and computational complexity by adding residual connections for skipping layers. It has been shown that consequently residual networks can reach significantly greater depths and achieve excellent classification accuracy.

We used a time series ResNet implementation obtained from Fawaz et al.'s repository (Ismail Fawaz [48]). The architecture has a $t \times 12$ input matrix representing the 12-leads with a length $t = 339$ and consists of three residual blocks of convolutional layers with filter sizes 64, 128, and 128, respectively. Each residual block contains three 1D convolutions with kernel sizes 8, 5, and 3, respectively. A global average pooling (GAP) and softmax output layer follow the residual blocks. In total, the neural network consists of three blocks and a depth of 11 layers (see Fig. 4).

The residual block architecture has been used in a variety of problem domains relating to time series classification, as studied by Ismail Fawaz et al. [14]. In both univariate and multivariate time series, ResNet outperformed other architectures across a majority of datasets. The only area where it did not—and rather underperformed significantly—was in ECG classification; the poor performance was attributed to the insufficient size of the dataset available for training the network. This brings about the need for data augmentation to increase the training dataset size (as well as balance the class sizes).

Signal Filtering and Beat Segmentation

In our work, the digitized ECG signal has been first filtered by a finite impulse response bandpass filter (0.05–35 Hz) to remove noise and baseline drift and, then, divided into a sequence of heartbeat segments. The resultant beat segments are used as the units of ECG data alteration and classification. Beat annotations in the ECG dataset are marked at the *R* peaks. By exploiting this R-peak marking, a dynamic segmentation process used by Veeravalli et al. [49] and Lin et al. [50] is used to extract ECG beat segments from the ECG time series; that is, a segment boundary is determined based on the previous RR interval, using the following formula.

$$P_{\text{window}} = \text{QR}_{\max} + 0.2 \times \text{RR}_{\max} + 0.1 \quad (1)$$

$$T_{\text{window}} = 1.5 \times \text{QT}_{c_{\max}} \times \sqrt{\text{RR}_{\text{prev}}} - \text{QR}_{\max} \quad (2)$$

where $\text{QR}_{\max}$ is set to 0.08, equivalent to half the maximum QRS duration, and the corrected QT interval, $\text{QT}_{c_{\max}}$, is set to 0.42. Conjoined P_{window} and T_{window} forms an ECG beat segment. Additionally, zero padding is done as needed to ensure that all segments have the same length.

Figure 5 shows an example ECG beat segment extracted using the formula, for one RBBB heartbeat. In the interest of space, only one lead (V1) is shown in Fig. 5a and a "complex-lead" combining the 12 leads is shown in Fig. 5b, deferring the plots of all individual 12 leads to the supplement (Boynton [29]). The lead V1 is clinically known to demonstrate an RBBB pattern well. The amplitude of a complex-lead ECG sample, y_i, is calculated as the arithmetic average of the differentials (i.e., approximate slopes) of the 12 lead ECG samples, that is, $y_i = \frac{1}{12} \sum_{j=1}^{12} |x_{j,i+1} - x_{j,i-1}|$ (Christov [51]). The gray dot marks the *R* peak; on its left is the *P* window, and on its right is the *T* window.

Data Augmentation

The augmentation procedure can be outlined in two steps—*peak detection* (Algorithm 1) and *peak alteration* (Algorithm 2), applied to the 12 channels of an ECG segment (each ECG lead gives a channel).

In the peak detection step, given the maximum number *N* of peaks, *N* most prominent peaks are found and returned from each channel's ECG segment. Specifically, the algorithm works as follows (see Algorithm 1). For each channel in the 12 lead ECG segment, first, characteristic peaks are found using the SciPy find_peaks function (Virtanen et al. [52]) (Line 5); a characteristic peak is defined as a local maximum point of at least the mean height of the samples in the input ECG. Then, the peaks found are ranked by their height by the SciPy peak_prominences function and are stored in the array peaksFound sorted by its prominence (Lines 6 and 7). Then, if the number of found peaks is equal to *N*, they are all added to the channel's peaks (Line 9), and if more than *N*, only the *N* most prominent peaks are selected and added (Line 12); otherwise (i.e., less than *N*), the peak finding step is repeated while reducing height threshold incrementally (i.e., by 0.01) until *N* peaks are found, which are then added (Lines 15–19). Finally, all peaks added for all 12 channels are returned (Line 21).

Algorithm 1 Amplitude-based ECG peak detection.

```
1: procedure FindPeaks(12leadECGsegment, N)
2: // Locate N peaks in the 12-lead ECG segment.
3:     for each channel ∈ 12leadECGsegment do
4:         height ← mean(channel) + 0.001          ▷ Height threshold for the peak-finding
5:         peaksFound ← find_peaks(channel, height)
6:         peaksProms ← peak_prominences(peaksFound)
7:         sort peaksFound by peaksProms
8:         if len(peaksFound) = N then
9:             peaks[channel] ← peaksFound
10:        end if
11:        if len(peaksFound) > N then
12:            peaks[channel] ← last N peaks in peaksFound    ▷ N most prominent peaks
13:        end if
14:        // Reduce the height threshold incrementally until N peaks are found.
15:        repeat
16:            height ← height − 0.01                ▷ Reduce the height threshold.
17:            peaksFound ← find_peaks(channel,height)
18:        until len(peaksFound) = N
19:        peaks[channel] ← peaksFound
20:    end for
21:    return peaks
22: end procedure
```

In the peak alteration step, each peak found from the 12-lead ECG segment is selected for alteration with the probability ρ, and, if selected, its amplitude is scaled by a factor f randomly selected between f_{min} and f_{max} and then smoothed using a smoothing function $\mathcal{F}_S$ (see Algorithm 2). Note that the peak times are not changed in our augmentation method to maintain the original rhythm of beats and that for each peak the same amplitude scaling is applied equally across all 12 channels to maintain their relative relationships in the original 12-lead ECG signal. To this end, the only peaks in a channel that are aligned with the corresponding peaks in the first channel (of the lead I) within sample distance δ are considered for amplitude scaling. Note as well that different ECG segments are scaled by different factors, as a new value of f is randomly selected every time Algorithm 2 is executed; this adds more diversity in the altered peaks across the segments, which we believe results in a more generalizable model than using one fixed factor for all peaks in the entire ECG record timeline.

The scaling and smoothing in the peak alteration work as follows. For each peak selected (Lines 3 and 4), first the scaling factor f for the peak is determined randomly in the given range. Then, for each channel, the baseline is determined by averaging the sample amplitudes within the channel (Line 8), and, if the peak is close enough (within δ sample distance) to the corresponding peak in the lead I's channel, the start point and the end point of the peak are identified as the points crossing the baseline upward and downward, respectively (Lines 12 and 13). Individual samples in the peak are then scaled by the factor f (Line 15). Next, the polarity of the peak is set to either positive, 1, or negative, -1, depending on whether the peak is located above or below the baseline (Lines 16–20), and the sign is given to the smoothing function (Line 23) to maintain the polarity. The scaled samples in the peak are then smoothed by the smoothing factors calculated by the smoothing function $\mathcal{F}_s$ (Lines 21–25).

Algorithm 2 Amplitude-based ECG peak alteration.

```
1: procedure AlterPeaks(12leadECGsegment, peaks, ρ, f_min, f_max, δ, F_S)
2: // With probability ρ, alter each peak's amplitude in the 12-lead ECG segment by a
   random scaling factor f ∈ [f_min, f_max] and the smoothing function F_S.
3:     for n = 1 to N do                          ▷ for each peak, where N = |peaks|
4:         With probability ρ do
5:         begin
6:             Pick a random value of f in the range of [f_min, f_max].
7:             for each channel ∈ 12leadECGsegment do
8:                 baseline ← mean(channel)
9:                 // Determine whether the peak is similar to the peak in the first channel.
10:                if the nth peak is within distance δ from the nth peak in lead I then
11:                    // Delineate the peak in the channel.
12:                    peakStart ← index of the first sample crossing the baseline from
                                   the peak to the left.
13:                    peakEnd ← index of the first sample crossing the baseline from
                                 the peak to the right.
14:                    // Apply scaling followed by smoothing to the peak.
15:                    scaledChannelPeak = f ∗ channel[peakStart : peakEnd]
16:                    if peak amplitude > baseline then
17:                        sign ← 1
18:                    else
19:                        sign ← −1
20:                    end if
21:                    smoothingFactor ← [ ]
22:                    for each sample ∈ channel[peakStart : peakEnd] do
23:                        append F_S(sample, variance(channel), sign) to smoothingFactor
24:                    end for
25:                    channel[peakStart:peakEnd] ← scaledChannelPeak ⊙ smoothingFactor
                                   ▷ Pairwise multiplication of array elements.
26:                end if
27:            end for
28:        end
29:    end for
30:    return 12leadECG                           ▷ with altered peaks
31: end procedure
```

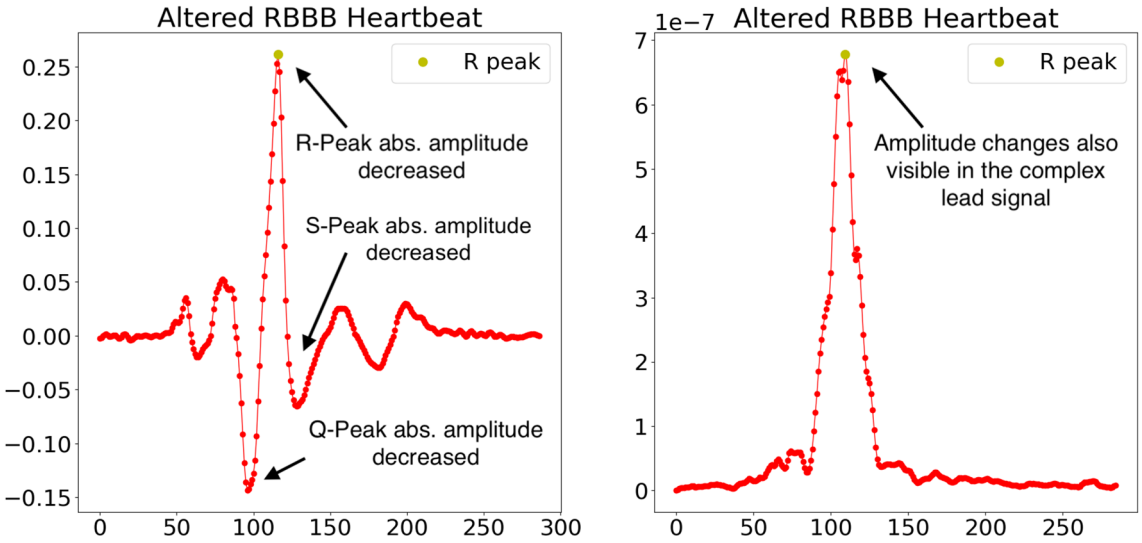
In our work, we set the number of peaks, N, to 5 to reflect the five peaks (i.e., P, Q, R, S, T) in a typical ECG segment; the probability of peak alteration, ρ, to 0.6 which is neutral with slight inclination for alteration; the range of scaling factor, ($f_{\min}$,$f_{\max}$), to (0.5, 1.5) determined through visual examination of altered ECG signals with Dr. Lustgarten, an electrophysiologist); and the sample distance δ between peaks of a channel and the lead I channel to 20 samples based on visual observations. Additionally, we used the Gaussian smoothing function $\mathcal{F}_S(x, y, z) = \frac{z}{y\sqrt{2\pi}} * e^{-\frac{x^2}{2y^2}}$. The output of $\mathcal{F}_S(x, y, z) \in [0, 1]$ is a smoothing factor, and the smoothed sample x is calculated as $x * \mathcal{F}_S(x, y, z)$ (see Line 25 of Algorithm 2).

Figure 6 shows an example of an ECG segment altered by applying Algorithm 1 and Algorithm 2 on the segment in Fig. 5—the lead V1 in Fig. 6a and the complex lead combining the 12 leads in Fig. 6b. The full 12-lead altered ECG segments can be seen in the supplement (Boynton [29]). Note the amplitudes of Q, R, and S peaks have been decreased as a random choice in this segment.

Given this data augmentation scheme, class size balancing is achieved by augmenting classes with lower representation (i.e., smaller number of beats) more and classes with greater representation less. So, the population size of the Normal class, the largest among all classes, is used as the baseline to determine the number of heartbeats to be augmented for each of the other classes to balance the size across all classes. ECG records of different patients are not distinguished and handled altogether.

Test-Time Augmentation

Conventionally, data augmentation is done on the training dataset, and testing is done using the original (non-augmented) dataset; in this case, however, excessively small classes in the imbalanced class distribution degrade the resulting test accuracy when averaged over all classes. To address this problem, test-time augmentation (TTA), used in image classification with good results (Moshkov et al. [53], Simonyan and Zisserman [54]), has been adopted for



(a) One-lead ECG (lead V1). (b) A complex-lead ECG.

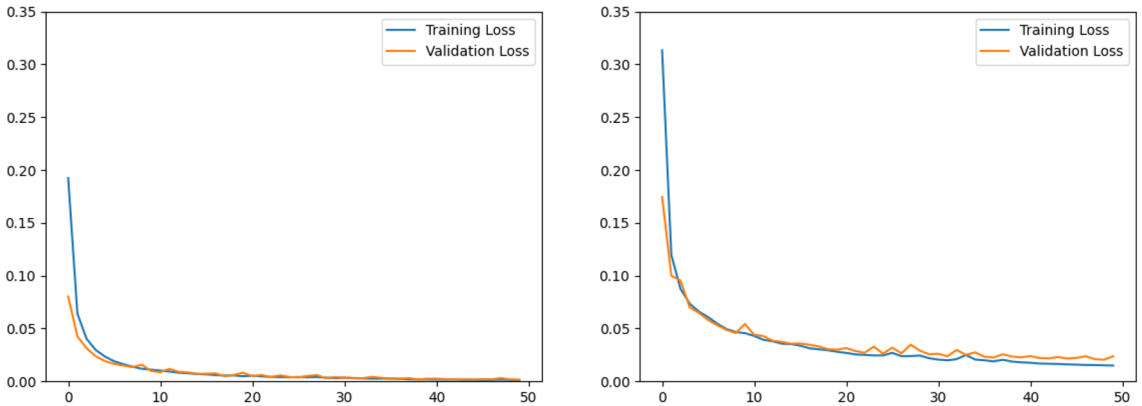
Fig. 6 A sample augmented ECG segment. (See the supplement (Boynton [29]) for the full 12-lead altered segments)

ECG time series segment classification in this work. TTA generates multiple augmented copies of each data item (i.e., ECG beats) in the test dataset, using the same data augmentation scheme used for the training dataset. Then, for each original beat, the classification accuracy is averaged over all augmented beats of the original one. Thus, TTA-based testing also evaluates the robustness and generalizability of the model.

Class Activation Map

The class activation map (CAM) was first introduced by Zhou et al. [25] to enable visual interpretation of convolutional models between input data and output classes (without additional parameters or modifications to the network architecture) by highlighting regions of the input data according to the level of contribution to the classification (Selvaraju et al. [55]). Wang et al. [13] soon applied CAM to time series data in their work on time series classification using ResNet. When applied to ECG, CAM would highlight important "regions" (i.e., subsequences) and patterns in the time series that contributed to the classification of a heartbeat.

In this work, we adapted the method used by Selvaraju et al. [55] for the one-dimensional convolutions in the time series ResNet. Specifically, the activation map for a time series is obtained by computing the gradient of the output y_c (before softmax) with respect to the activation of the last



(a) EXP 1 and EXP 2. (b) EXP 3 and EXP 4.

Fig. 7 Loss curves (average training loss and validation loss in 10-fold cross validation) during model training. (The plot is truncated to 50 epochs)

Table 4 Classification accuracy by the diagnostic class in each of the four data augmentation scenarios

Diagnostic class	Data augmentation scenario	
	EXP1	EXP2
	Se / Sp / Pr / BA / F1	Se / Sp / Pr / BA / F1
NORMAL	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 1.00 / 1.00 / 1.00
PVC	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 1.00 / 1.00 / **1.00**
RBBB	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 1.00 / 1.00 / 1.00
APC	1.00 / 1.00 / 1.00 / 1.00 / 1.00	0.98 / 1.00 / 0.97 / 0.98 / 0.98
FUSION	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 0.93 / 0.96 / 0.96
NESC	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 1.00 / 1.00 / 1.00
SVESC	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 0.59 / 0.80 / 0.74
SVPB	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 0.50 / 0.75 / 0.67
UNKNOWN	1.00 / 1.00 / 1.00 / 1.00 / 1.00	1.00 / 1.00 / 1.00 / 1.00 / 1.00
Class-balanced accuracy	1.00	1.00
Macro-average F1	1.00	0.93
Class-balanced accuracy	1.00	1.00
Macro-average F1	1.00	0.93
Diagnostic class	Data augmentation scenario	
	EXP3	EXP4
	Se / Sp / Pr / BA / F1	Se / Sp / Pr / BA / F1
NORMAL	1.00 / 0.59 / 0.23 / 0.62 / 0.38	1.00 / 0.98 / 1.00 / 1.00 / 1.00
PVC	0.76 / 0.89 / 0.55 / 0.66 / 0.64	1.00 / 1.00 / 1.00 / 1.00 / 1.00
RBBB	1.00 / 1.00 / 0.99 / 1.00 / 0.99	0.99 / 1.00 / 1.00 / 1.00 / 1.00
APC	0.99 / 1.00 / 0.96 / 0.98 / 0.98	0.81 / 1.00 / 0.96 / 0.94 / 0.88
FUSION	0.47 / 1.00 / 0.99 / 0.73 / 0.64	0.43 / 1.00 / 0.95 / 0.93 / 0.59
NESC	0.66 / 1.00 / 1.00 / 0.83 / 0.80	0.92 / 1.00 / 1.00 / 0.92 / 0.96
SVESC	0.00 / 1.00 / 1.00 / 0.50 / 0.30	0.00 / 1.00 / N/A / 0.00 / 0.00
SVPB	0.00 / 1.00 / N/A / 0.00 / 0.00	0.00 / 1.00 / N/A / 0.00 / **0.00**
UNKNOWN	0.00 / 1.00 / N/A / 0.00 / 0.00	0.00 / 1.00 / N/A / 0.00 / 0.00
Class-balanced accuracy	0.54	0.57
Macro-average F1	0.48	0.60
Class-balanced accuracy	0.54	0.57
Macro-average F1	0.48	0.60

EXP1 training and testing, *EXP2* training only, *EXP3* testing only, *EXP4* neither, *Se* sensitivity (recall), *Sp* specificity, *Pr* precision, *BA* balanced accuracy, *F1* F1 score

N/A means there is no true positive for the class and, therefore, the Precision is undefined

convolutional layer A^k, that is, $\frac{\partial y_c}{\partial A^k}$. The importance of a filter to the classification, w_k^c, is then given by averaging the gradients as

$$w_k^c = \frac{1}{Z} \sum_i \frac{\partial y_c}{\partial A_i^k}. \tag{3}$$

and the activation map M indicating the importance of a subsequence is built using the activation function ReLU as

$$M = ReLU(\sum_k w_k^c A^k(t)). \tag{4}$$

Evaluations

Experiment Setup

Data Augmentation Cases

In view of handling class imbalances through data augmentation, the evaluation considers four scenarios depending on whether each of the training dataset and test dataset is augmented or not (i.e., original).

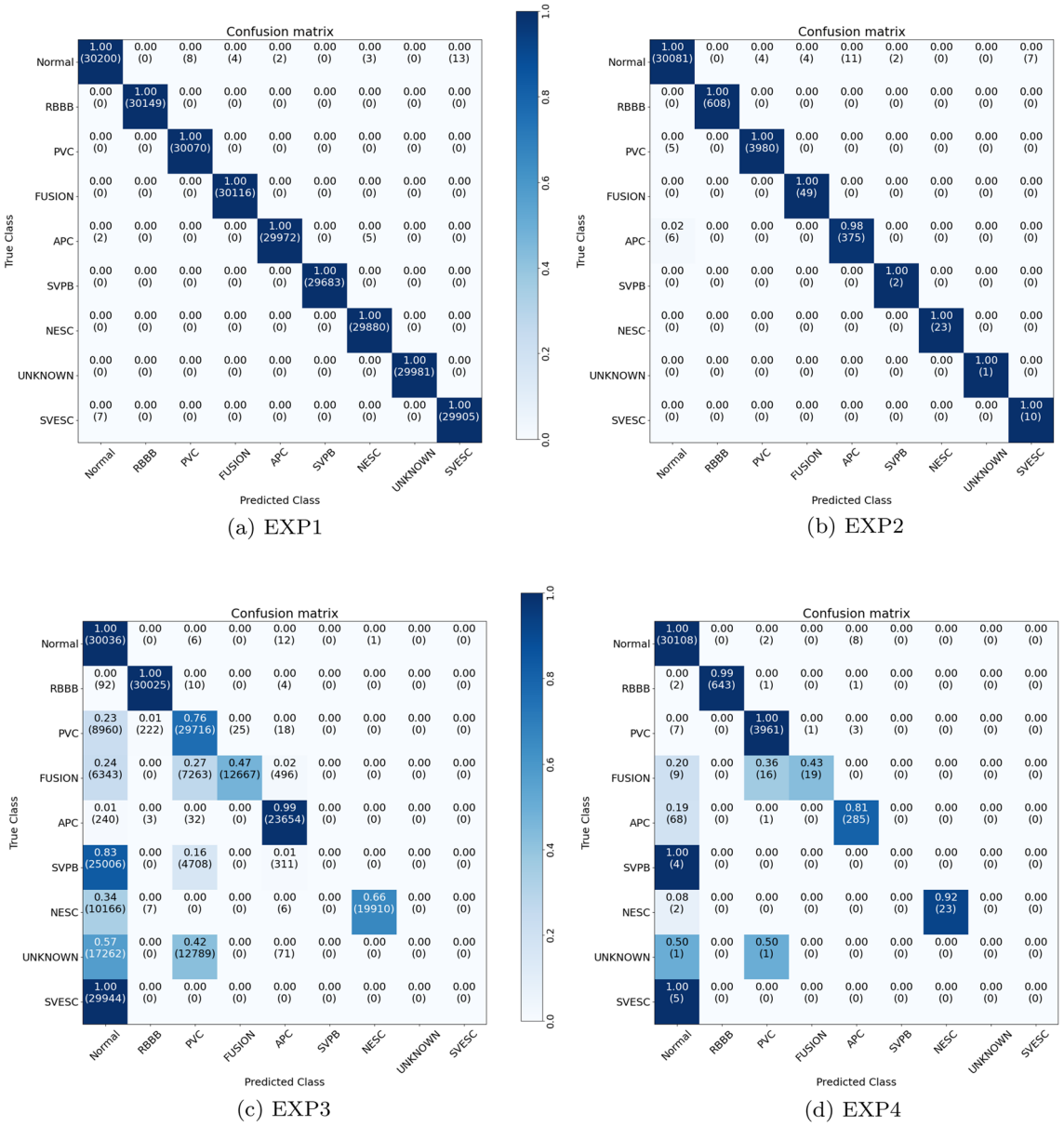


Fig. 8 Confusion matrices in the four data-augmentation scenarios. Each entry shows the ratio of the true class beats over the predicted class beats (top) and the predicted class size (bottom). The shade indicates the accuracy—darker for higher accuracy

EXP1 Augmented training and augmented testing.
EXP2 Augmented training and original testing.
EXP3 Original training and augmented testing.
EXP4 Original training and original testing.

Performance Metrics

Classification accuracy metrics commonly used in health informatics and information retrieval are used comprehensively, that is, the three basic metrics—sensitivity (= recall), specificity, and precision---and the two composite metrics---balanced accuracy (= (sensitivity + specificity)/2) and F1 score (= 2*precision*recall/(precision + recall)). Further, to account for the effect of class imbalances in the original (i.e., not augmented) training/test datasets, an *adjusted class-balanced accuracy* (Pedregosa et al. [56]) is used. Class-balanced accuracy is the macro-average of sensitivity (= recall) scores per class. This accuracy is then adjusted for randomness by subtracting the accuracy achieved if the class predictions were completely random, which would be $1/n$ where n is the number of classes. This adjusted class-balanced accuracy would be 0 for completely random class predictions and 1 for completely correct class predictions. In

addition, a *macro-average F1* score, computed as the arithmetic average of the F1 scores per class, is used as well.

Training Scheme

The entire ECG record dataset has been split into a training dataset and a test dataset by a 80–20% split. Training was done using stochastic gradient descent and stratified *k*-fold cross validation. The stratification was to cope with class imbalances of the original (i.e., not augmented) datasets in the EXP2, EXP3, and EXP4 scenarios. The termination condition was the validation loss change being no more than 10^{-4} for 10 consecutive epochs. The learning rate was initially set to 0.01 and reduced to half each time following four epochs where there was no improvement in the validation loss with a minimum learning rate of 10^{-6}.

Figure 7 shows the training and validation losses during ten-fold cross validation training. With the augmented

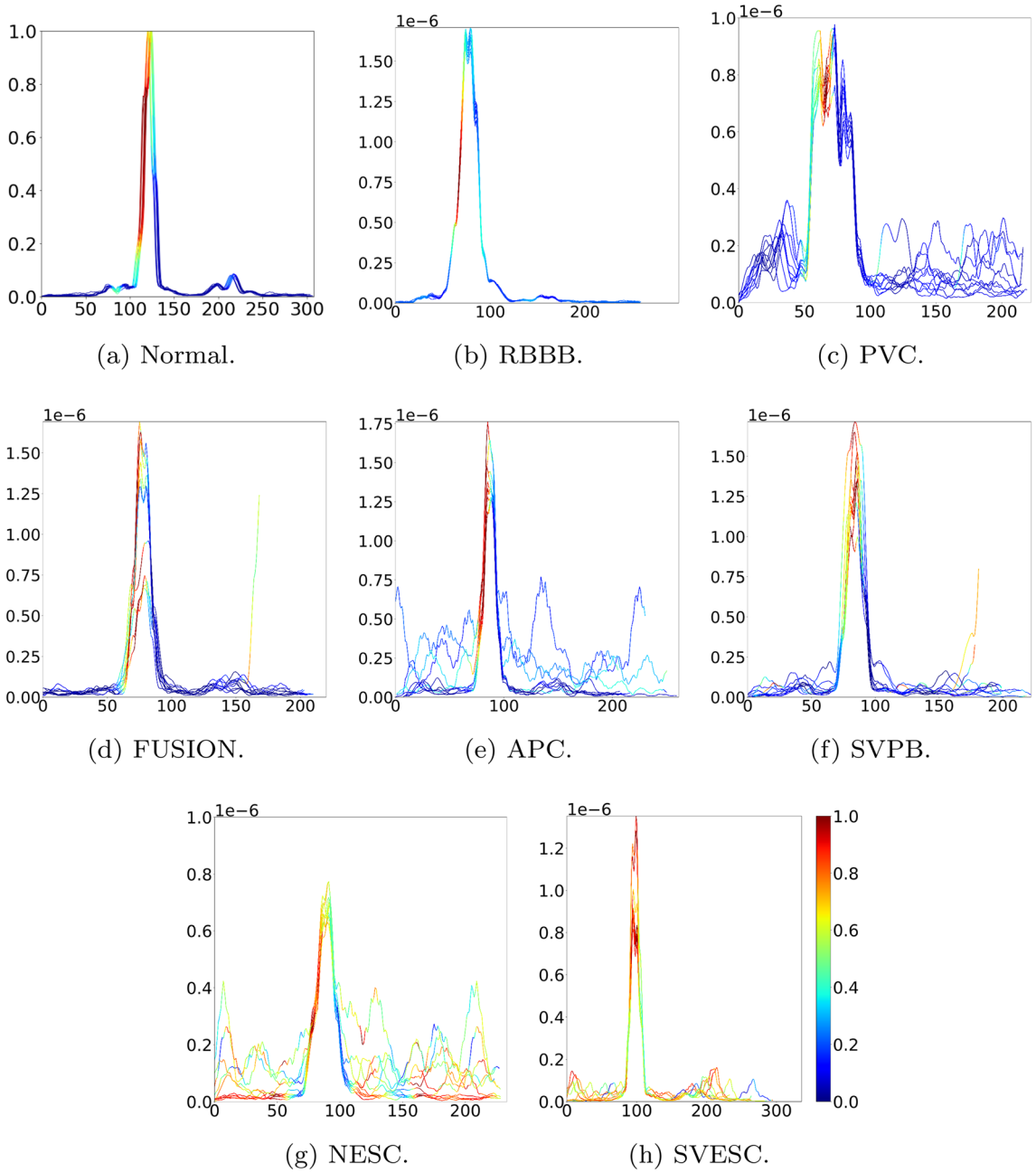


Fig. 9 Heatmap patterns of beats correctly classified by the model trained with augmented dataset (in EXP1 and EXP2). Test beats are from the original dataset

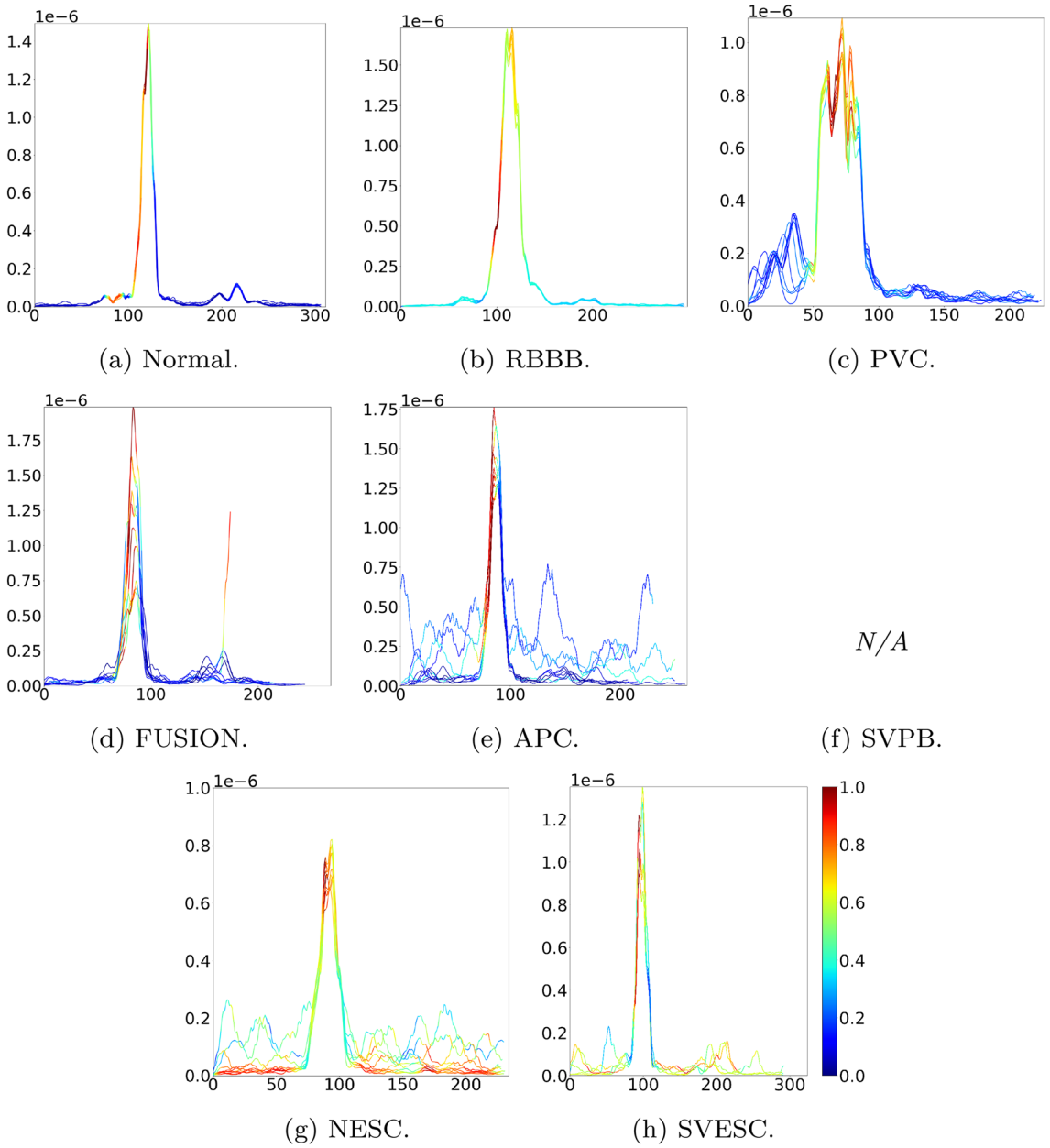


Fig. 10 Heatmap patterns of beats correctly classified by the model trained with non-augmented dataset (in EXP3 and EXP4). Test beats are from the original dataset. There is no SVBP beat correctly classified. There are only five correctly classified SVESC beats

(hence larger) training dataset (Fig. 7a), the average number of epochs until convergence was 14 and the ratio of validation to training losses was 1.12. On the other hand, with the non-augmented (hence much smaller) training dataset (Fig. 7b), the convergence epoch was 23 and the loss ratio was 1.17 but the loss ratio increased throughout training until epoch 74. Evidently, the smaller size of dataset resulted in more overfitting.

Results and Discussions

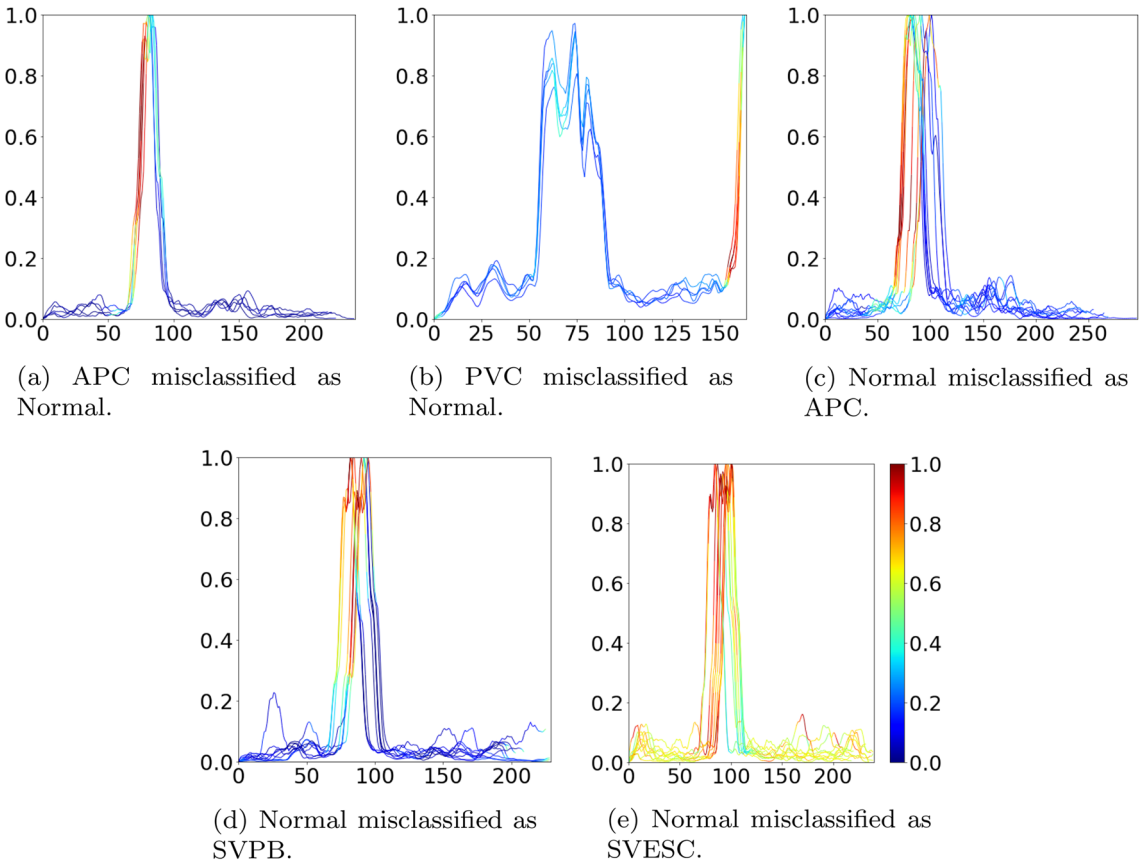
Classification Accuracy

Table 4 summarizes the classification accuracy in each of the four augmentation scenarios, and Fig. 8 shows the confusion matrix in each scenario. The following observations are made from these results.

First, augmentation of the dataset does help achieve higher accuracy. Considering both the class balanced accuracy (CBA) and the macro-average F1 (MAF) score, the order of overall classification accuracy from the highest first is EXP1, EXP2, EXP4, and EXP3, which is in the order of both training and testing augmented (EXP1), only training augmented (EXP2), neither training nor testing augmented (EXP4), and only testing augmented (EXP3). So, the best is to augment both training and testing datasets and the worst is to augment the testing dataset without augmenting the training dataset.

Second, augmentation of the training dataset plays a major role in achieving the accuracy. Note EXP1 and EXP2 use an augmented training dataset while EXP3 and EXP4 do not, and the accuracy drop from EXP1 to EXP3 is 46% for CBA and 52% for MAF and from EXP2 to EXP4 is 43% for CBA and 33% for MAF. In comparison, augmentation of the test dataset plays a relatively minor role. For instance, the accuracy drop from EXP1 to EXP2 (where EXP1 uses an augmented test dataset and EXP2 does not) is 0% for CBA and 7% for MAF, while the accuracy rise from EXP3 to EXP4 (where EXP3 uses an augmented test dataset and EXP4 does not) is 3% for CBA and 12% for MAF. This difference between training dataset and test dataset makes sense—the training dataset is larger than the test dataset (by the ratio of 80–20% in this work) and more diverse in samples than the test dataset and, therefore, has a larger effect on the learning performance.

Third, while augmentation of the test dataset helps improve the accuracy overall, it is not the case for every class unless the training dataset is augmented as well. Between EXP1 and EXP2, the F1 (and the associated precision) score is higher for EXP1 than for EXP2 in four smaller classes (i.e. APC, FUSION, SVESC, SVPB) while the same for three larger classes (i.e., Normal, PVC, RBBB). This indicates that once the training dataset is augmented, augmenting the test dataset further improves the accuracy, especially for small classes. Between EXP3 and EXP4, however, the relative classification accuracy differs depending on the class size. That is, EXP3 tends to show lower accuracy than EXP4 for larger classes (i.e., Normal, PVC, RBBB) and higher accuracy than EXP4 for smaller classes (i.e., APC, FUSION, SVESC). This result indicates that when the training dataset is not augmented, augmenting the test dataset may compromise the robustness of the model because it exposes the model to features that it has not learned during training,



(a) APC misclassified as Normal.

(b) PVC misclassified as Normal.

(c) Normal misclassified as APC.

(d) Normal misclassified as SVPB.

(e) Normal misclassified as SVESC.

Fig. 11 Heatmaps of five cases of misclassification (in EXP2)

thereby leading to a biased distribution of false positives and false negatives for different classes [see "result_details" in the supplement (Boynton [29])]. Thus, overall it is important to augment both training and test datasets.

These observations affirm that the near 100% classification accuracy achieved is attributed more to the data augmentation with class balancing than the ResNet deep learning alone. As shown by Ismail Fawaz et al. [14], ResNet is one of the best performers for time series data and ResNet has demonstrated outstanding classification performance for a variety of tasks (He et al. [12], Labati et al. [10], Ardakani et al. [57]. It, however, performed relatively poor (by almost 52%) for the task of classifying 12-lead ECGs into nine classes when trained without any training data augmentation (i.e., as in EXP3 and EXP4) as opposed to with (as in EXP1 and EXP2).

Classification Signatures

Heatmap patterns [(obtained through the class activation map ("Class Activation Map"))] can provide a good visual clue associating a region of heartbeat and the diagnostic class of the beat. A complex signal (e.g., vector of 12 lead channels) was adequate for use to identify the pattern, as all 12 channels showed consistent heatmap patterns across beats.

Figure 9 shows such heatmap patterns in each diagnostic class (except Unknown) from the model trained with augmented dataset in EXP1 and EXP2, and Fig. 10 shows those from the model trained with non-augmented datasets in EXP3 and EXP4. Each heatmap plot is a superimposition of ten randomly selected beats. The same beats have been used between the two cases (Figs. 9 and 10). The resulting heatmap patterns were consistent across the selected beats. Each plot clearly shows the "hot" region that is important to recognizing the class; let us call such a pattern the "classification signature". For example, the hot segment rising from the end of the *Q* wave to the *R* peak in Fig. 9a is a signature of the Normal class, and the hot segment rising from one-third to two-third of the rising *R* wave in Fig. 9b is a signature of the RBBB class.

Comparing the heatmaps between the cases of augmented training dataset (see Fig. 9) and non-augmented training dataset (see Fig. 10) shows that the heatmaps are less clear in the latter case. Evidently, this is attributed to the poorer model accuracy resulting from non-augmented training dataset (see Table 4). For example, NESC in Fig. 10g shows red in a much larger region than in Fig. 9g; Normal in Fig. 10a shows lighter red rising edge of the *R* wave than in Fig. 9a; FUSION in Fig. 10d shows more red in the rising and falling edges of the *R* wave than in Fig. 9d.

As an anecdotal example, let us examine the five most common cases of misclassification observed in the confusion matrix of EXP2 (see Fig. 8b)—(1) APC falsely classified as Normal; (2) PVC falsely classified as Normal; (3) Normal as SVESC; (4) Normal as APC; (5) Normal as SVPB. (The first two, false normal, cases would be of particular importance in cardiac care). Figure 11 shows the heatmaps of these five cases. Comparison with the heatmaps of correctly classified beats (see Fig. 9) demonstrates the effectiveness of the heatmaps as an indicator of the class, as discussed in the following observations.

First, APC beats misclassified as Normal beats (Fig. 11a) are much noisier in the entire beat comprising *P* wave, QRS complex, and *T* wave than correctly classified APC beats. Note that both Normal beats and APC beats have the hot segment on the rising edge of the *R* wave. So, it appears the noise has misled the model to mistake it for a Normal beat.

Second, PVC beats misclassified as Normal (Fig. 11b) have a rising edge at the end of the segment, apparently included in the segment due to shorter *T* windows of the PVC beats showing an early (i.e., premature) QRS complex, consequently causing an anomaly in the data segmentation (discussed in "Signal Itering and Beat Segmentation"). The rising edge looks like the one in the Normal beats and is suspected to cause the misclassification. (To verify this, we manually trimmed out the false rising edge and reclassified it, and confirmed correct classification to PVC.)

Third, as for the Normal beats misclassified as APC, SVPB, and SVESC beats (Figs. 11c–e), their heatmaps indeed show rising edges of the *R* wave that look more similar to those in the misclassified class than the Normal class.

Conclusion

This paper presented the classification performance of deep learning (ResNet time series version) with a focus on resolving the class imbalance problem through an amplitude-alteration scheme for data augmentation. The results showed that augmenting the (larger) training dataset has a major benefit and augmenting the (smaller) test dataset has a relatively minor benefit and that the best is to augment both datasets. This paper also presented using the class activation map to identify heatmap signatures signaling different diagnostic classes. The consistency of heatmap patterns across the selected beats and the higher clarity of patterns from a model generated with an augmented dataset also ascertains the benefit of the proposed data augmentation in the time series ECG data. Overall, the results strongly suggest the effectiveness of the amplitude-altered ECG beat augmentation scheme.

There are a number of suggested further work. First, in this paper all 12 leads were always considered together, but it may reveal interesting results to consider only selected part of the 12 leads; this would lead to a computational feature

selection problem for identifying dominant leads. Second, the classification model in this paper was not personalized, that is, no distinction was made between different patients when ECG data were augmented. It is, however, commonly understood that personalized ECG analysis produces more accurate results (Lin et al [49]), and in this regard, personalized data augmentation can be studied. Third, this paper only introduced the idea of using the CAM as a potentially effective tool to facilitate the beat classification; this idea deserves more in-depth study toward eventually building a heatmap signature profile of diagnostic classes.

Acknowledgements The work by Edmund Do and Jack Boynton was supported by the College of Engineering and Mathematical Sciences at the University of Vermont through the Research Experience for Undergraduates program. Computations were performed on the Vermont Advanced Computing Core supported in part by NSF award No. OAC-1827314. The authors thank the anonymous reviewers for their comments, which were invaluable for improving the quality of the manuscript.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. da Luz EJS, Schwartz WR, Cámara-Chávez G, Menotti D. ECG-based heartbeat classification for arrhythmia detection: a survey. Comput Methods Programs Biomed. 2016;127:144–64.
2. Chawla NV, Japkowicz N, Kotcz A. Special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl. 2004;6(1):1–6.
3. Hwang B, You J, Vaessen T, Myin-Germeys I, Park C, Zhang BT. Deep ECGnet: an optimal deep learning framework for monitoring mental stress using ultra short-term ECG signals. Telemed e-Health. 2018;24(10):753–72.
4. Manogaran G, Shakeel PM, Fouad H, Nam Y, Baskar S, Chilamkurti N, Sundarasekar R. Wearable iot smart-log patch: an edge computing-based bayesian deep learning network system for multi access physical monitoring system. Sensors. 2019;19(13):3030.
5. Trenta F, Conoci S, Rundo F, Battiato S. Advanced motion-tracking system with multi-layers deep learning framework for innovative car-driver drowsiness monitoring. In: Proceedings of the 14th IEEE international conference on automatic face gesture recognition (FG 2019). 2019; pp. 1–5.
6. Jm Kwon, Cho Y, Jeon KH, Cho S, Kim KH, Baek SD, Jeung S, Park J, Oh BH. A deep learning algorithm to detect anaemia with ECGs: a retrospective, multicentre study. Lancet Digit Health. 2020;2(7):e358–67.
7. Han C, Shi L. ML-ResNet: a novel network to detect and locate myocardial infarction using 12 leads ECG. Comput Methods Programs Biomed. 2020;185:105138.
8. He R, Liu Y, Wang K, Zhao N, Yuan Y, Li Q, Zhang H. Automatic cardiac arrhythmia classification using combination of deep residual network and bidirectional LSTM. IEEE Access. 2019;7:102119–35.
9. Tung H, Zheng C, Mao X, Qian D. Multi-lead ECG classification via an information-based attention convolutional neural network. CoRR. arXiv:2003.12009, 2020.
10. Labati RD, Muñoz E, Piuri V, Sassi R, Scotti F. Deep-ECG: convolutional neural networks for ECG biometric recognition. Pattern Recognit Lett. 2019;126:78–85.
11. Aurore L, Ana M, Pablo MJ, Pablo L, Blanca R. Computational techniques for ECG analysis and interpretation in light of their contribution to medical advances. J R Soc Interface. 2018;15(138):20170821.
12. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 770–778.
13. 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). 2017; pp. 1578–1585.
14. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. Data Min Knowl Discov. 2019;33(4):917–63.
15. PhysioNet. PysioBank annotation. 2016. https://archive.physionet.org/physiobank/annotations.shtml. Accessed 6 Dec 2020.
16. Alday EAP, Gu A, Shah A, Liu C, Sharma A, Seyedi S, Rad AB, Reyna M, Clifford G. Classification of 12-lead ECGs: the PhysioNet—computing in cardiology challenge 2020. 2020. https://physionet.org/content/challenge-2020/1.0.1/. Accessed 6 Dec 2020.
17. Pan Q, Li X, Fang L. Data augmentation for deep learning-based ECG analysis. In: Liu C, Li J, editors. Feature engineering and computational intelligence in ECG monitoring. Singapore: Springer; 2020. p. 91–111.
18. Cui Z, Chen W, Chen Y. Multi-scale convolutional neural networks for time series classification. arXiv preprint. 2016. arXiv:160306995.
19. Um TT, Pfister F, 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; 2017.
20. Cao P, Li X, Mao K, Lu F, Ning G, Fang L, Pan Q. A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation. Biomed Signal Process Control. 2020;56:101675.
21. Le Guennec A, Malinowski S, Tavenard R. Data augmentation for time series classification using convolutional neural networks. In: Proceedings of the ECML/PKDD workshop on advanced analytics and learning on temporal data, 2016.
22. Jun TJ, Nguyen HM, Kang D, Kim D, Kim D, Kim YH. ECG arrhythmia classification using a 2-D convolutional neural network. ArXiv. 2018. arXiv:1804.06812.
23. Yao Q, Wang R, Fan X, Liu J, Li Y. Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network. Inf Fusion. 2020;53:174–82.
24. Acharya U, Oh SL, Hagiwara Y, Tan J, Adam M, Gertych A, Tan R. A deep convolutional neural network model to classify heartbeats. Comput Biol Med. 2017;89:389–96.
25. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR). 2016; pp. 2921–2929.
26. Goodfellow SD, Goodwin A, Greer R, Laussen PC, Mazwi M, Eytan D. Towards understanding ECG rhythm classification using convolutional neural networks and attention mappings. In: Proceedings of the machine learning for healthcare conference, PMLR, 2018; pp. 83–101.

27. Oh SL, Ng EY, San Tan R, Acharya UR. Automated beat-wise arrhythmia diagnosis using modified U-net on extended electrocardiographic recordings with heterogeneous arrhythmia types. Comput Biol Med. 2019;105:92–101.
28. Wang J, Qiao X, Liu C, Wang X, Liu Y, Yao L, Zhang H. Automated ECG classification using a non-local convolutional block attention module. Comput Methods Programs Biomed. 2021;203:106006.
29. Boynton J. 12-Lead imbalanced ECG beat classification using time series ResNet. 2020. https://github.com/JackWBoynton/ECG_classification_ResNet/. Accessed 6 Dec 2020.
30. Rajesh KN, Dhuli R. Classification of ECG heartbeats using nonlinear decomposition methods and support vector machine. Comput Biol Med. 2017;87:271–84.
31. Rahman QA, Tereshchenko LG, Kongkatong M, Abraham T, Abraham MR, Shatkay H. Utilizing ECG-based heartbeat classification for hypertrophic cardiomyopathy identification. IEEE Trans Nano Biosci. 2015;14(5):505–12.
32. Liang W, Zhang Y, Tan J, Li Y. A novel approach to ECG classification based upon two-layered HMMS in body sensor networks. Sensors. 2014;14(4):5994–6011.
33. Lyon A, Mincholé A, Martínez JP, Laguna P, Rodriguez B. Computational techniques for ECG analysis and interpretation in light of their contribution to medical advances. J R Soc Interface. 2018;15(138):20170821.
34. Li Y, Pang Y, Wang J, Li X. Patient-specific ECG classification by deeper CNN from generic to dedicated. Neurocomputing. 2018;314:336–46.
35. Yildirim Özal. A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. Comput Biol Med. 2018;96:189–202.
36. Saadatnejad S, Oveisi M, Hashemi M. LSTM-based ECG classification for continuous monitoring on personal wearable devices. IEEE J Biomed Health Inform. 2020;24(2):515–23.
37. Hong S, Zhou Y, Shang J, Xiao C, Sun J. Opportunities and challenges of deep learning methods for electrocardiogram data: a systematic review. Comput Biol Med. 2020;122:103801.
38. Liu F, Zhou X, Cao J, Wang Z, Wang H, Zhang Y. A LSTM and CNN based assemble neural network framework for arrhythmias classification. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2019; pp. 1303–1307.
39. Lynn HM, Pan SB, Kim P. A deep bidirectional GRU network model for biometric electrocardiogram classification based on recurrent neural networks. IEEE Access. 2019;7:145395–405.
40. Zhou Y, Zhang H, Li Y, Ning G. ECG heartbeat classification based on ResNet and Bi-LSTM. IOP Conf Ser Earth Environ Sci. 2020;428:012014.
41. Moody GB, Mark RG. MIT-BIH arrhythmia database. 1992. https://physionet.org/content/mitdb/. Accessed 6 Dec 2020.
42. American National Standard. ANSI/AAMI. Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms. In: 1998-2008; ISO EC57. Technical report, association for the advancement of medical instrumentation. 2012. https://my.aami.org/aamiresources/previewfiles/EC57_1212_preview.pdf. Accessed 6 Dec 2020.
43. PhysioNet. St Petersburg INCART 12-lead arrhythmia database. 2008. https://physionet.org/content/incartdb/1.0.0/. Accessed 6 Dec 2020.
44. Randazzo A (2016) Guide to 12-lead ECG placement. https://www.primemedicaltraining.com/12-lead-ecg-placement/. Accessed 6 July 2021.
45. Lieberman K. Interpreting 12-lead ECGs: a piece by piece analysis. Nurse Pract. 2008;33(10):28–35.
46. Goldberger A, Amaral L, Glass L, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2003;101(23):e215–20.
47. Shimodaira H. Improving predictive inference under covariate shift by weighting the log-likelihood function. J Stat Plan Inference. 2000;90(2):227–44.
48. Fawaz HI. Deep learning for time series classification. 2020. https://github.com/hfawaz/dl-4-tsc. Accessed 1 Sept 2021.
49. Lin Y, Lee BS, Lustgarten D. Continuous detection of abnormal heartbeats from ECG using online outlier detection. In: Information management and big data. Berlin: Springer; 2019. p. 349–66.
50. Veeravalli B, Deepu CJ, Ngo D. Real-time, personalized anomaly detection in streaming data for wearable healthcare devices. In: Zomaya AY, Abbas A, Khan SU, editors. Handbook of large-scale distributed computing in smart healthcare. Berlin: Springer; 2017. p. 403–26.
51. Christov II. Real time electrocardiogram QRS detection using combined adaptive threshold. Biomed Eng Online. 2004;3:28.
52. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17:261–72.
53. Moshkov N, Mathe B, Kertesz-Farkas A, Hollandi R, Horvath P. Test-time augmentation for deep learning-based cell segmentation on microscopy images. Sci Rep. 2020;10(5068).
54. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the international conference on learning representations 2015, 2015.
55. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. Int J Comput Vis. 2019;128(2):336–59.
56. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
57. Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A. Application of deep learning technique to manage COVID-19 in routine clinical practice using ct images: results of 10 convolutional neural networks. Comput Biol Med. 2020;121:103795.