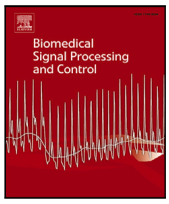
Biomedical Signal Processing and Control 112 (2026) 108469

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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



Generative deep learning and signal processing for data augmentation of cardiac auscultation signals: Improving model robustness using synthetic audio

Leigh Abbott, Milan Marocchi *, Matthew Fynn, Yue Rong, Sven Nordholm

School of Electrical Engineering, Computing, and Mathematical Sciences (EECMS), Faculty of Science and Engineering, Curtin University, Bentley, 6102, WA, Australia

ARTICLE INFO

Keywords:
Data augmentation
Denoising diffusion probabilistic models
Generative deep learning
Abnormal heart sound classification
Synthetic audio generation

ABSTRACT

Accurately interpreting cardiac auscultation signals is essential for diagnosing and managing cardiovascular diseases. However, the paucity of labelled data inhibits classification models' training. Researchers have turned to generative deep learning techniques alongside signal processing to augment existing data and improve cardiac auscultation classification models. However, the primary focus of prior studies has been on model performance rather than robustness. Robustness, in this case, is defined as both in-distribution and out-of-distribution performance by measures such as Matthew's correlation coefficient. One contribution of this work is to show that more robust abnormal heart sound classifiers can be trained using an augmented dataset. The augmented dataset includes both signal processing techniques and synthetic phonocardiograms conditionally generated using the WaveGrad and DiffWave diffusion models an approach that, to the best of our knowledge, is the first work of its kind. The efficacy of the proposed data augmentation approach is evaluated on an example convolutional neural network, trained on the original and augmented data. It is found that both the in-distribution and out-of-distribution performance can be improved over various datasets for neural networks trained with this augmented dataset. Results show significant performance improvements. Specifically, in-distribution accuracy, balanced accuracy, and Matthew's correlation coefficient (MCC) increased by 2.5%, 4.1%, and 0.066, respectively. The greatest out-of-distribution improvements were observed on one dataset, where accuracy, balanced accuracy, and MCC increased by 43.1%, 20.2%, and 0.297, respectively. These improvements across all metrics highlight that augmented datasets significantly address issues of imbalanced data, ultimately leading to more generalisable and robust classifiers.

1. Introduction

Cardiovascular disease (CVD) is the primary contributor to mortality worldwide, representing more than 30 % of all global deaths in 2019 [1]. In addition to the human cost, CVD places an immense economic burden on healthcare systems and society [1]. To treat CVD effectively, it is necessary to diagnose and evaluate the condition of the heart accurately.

Cardiac auscultation (CA) is the process of listening to sounds generated by the heart [2]. Physicians have traditionally performed CA using stethoscopes to detect and monitor heart conditions in a non-invasive manner. However, the difficulty of performing CA leads to uncertainty in diagnosis and poor patient outcomes. The issue is further complicated by the fact that CA is both difficult to teach and a specialised skill, with studies noting that primary care physicians often lack proficiency in this area [2].

Recently, a wearable multichannel electrophonocardiography (EPCG) device has been developed [3]. The premise of this device is to detect CVD utilising synchroised phonocardiogram (PCG) and electrocardiogram (ECG) data. The combination of these signals can result in more accurate and robust classifications. However, there is currently limited synchronised multichannel phonocardiogram and electrocardiogram (SMPECG) data, which creates a need for a technique to aid in creating a larger dataset.

There are current limitations that prevent robust classification results across multiple datasets. These include a lack of quality data and unbalanced datasets, with most data having lots of background noise, resulting in a low signal-to-noise ratio. There is also a limited amount of synchronised PCG and ECG recordings, which limits the effectiveness of algorithms, despite the large amounts of standalone

* Corresponding author.
E-mail address: milan.marocchi@postgrad.curtin.edu.au (M. Marocchi).

https://doi.org/10.1016/j.bspc.2025.108469
Received 28 January 2025; Received in revised form 17 June 2025; Accepted 2 August 2025

ECG and some PCG data. Traditional augmentation approaches can help to overcome these issues, with augmentation being applied to existing signals [4]. This is somewhat lacking, however, as it does not always increase the out-of-distribution performance, leaving room for further approaches to address this issue. With recent advancements in conditional waveform generation using diffusion models [5,6], it is possible to extend previously ECG-only datasets by generating PCG signals conditioned from the ECG in these datasets.

This work explores traditional augmentation approaches alongside the generation of synthetic signals, to create more robust classifiers of abnormal heart sounds.

The main contributions of this work are summarised below:

- Development of a diffusion model to create PCG signals conditional on existing ECG signals, allowing additional data to be used from ECG datasets once the diffusion model has created the corresponding PCG signal. To the best of our knowledge, this is the first work using diffusion models to generate PCG signals.
- Traditional augmentation methods synchronised over the PCG and ECG signals and extensive methods beyond those utilised in other studies.
- Augmentation methods were applied to a top-performing model [7] on the training-a dataset [8], resulting in improvements of 2.5% in accuracy, 4.1% in balanced accuracy, 1.9% in F_1^+ score, and 0.066 in Matthew's Correlation Coefficient (MCC). Additionally, when tested on the training-e dataset — where the model had not been trained on any of the dataset's data — there were notable improvements of 43.1% in accuracy, 20.2% in balanced accuracy, 27.1% in F_1^+ score, and 0.297 in MCC.

The remainder of the paper is organised as follows. Background in PCG and ECG signals is covered in Section 2. Literature survey on model robustness, biomedical signal augmentation and generative models is presented in Section 3. Following this, the methods and results are presented in Sections 4 and 5 before a discussion of the results in Section 6 and the final conclusions and further work are summarised in Section 7.

2. Background

2.1. Phonocardiogram and electrocardiogram signals

PCG signals comprise multiple sounds from the opening and closing of valves and blood flow inside the heart that cause vibrations, which are then recorded from the chest wall [9]. The fundamental heart sounds are the first (S1) and second (S2) sounds, which are the most prominent. The S1 occurs during the beginning of the systole and is caused by isovolumetric ventricular contraction. S2 is caused by the closing of the aortic and pulmonic valves during the beginning of the diastole. Although the S1 and S2 sounds are the most audible, PCG signals consist of many other heart sounds such as the third (S3) and fourth (S4) heart sounds, systolic ejection clicks, mid-systolic clicks, opening snap and heart murmurs [8]. These heart murmurs are produced by turbulent flowing blood, which can indicate the presence of particular CVDs. These various heart sounds all lie within the low frequencies, with S1 from 10 Hz–140 Hz and the highest energy around 25 Hz–45 Hz. The S2 is from 10 Hz–200 Hz, with most of the energy around 55 Hz–75 Hz. S3 and S4 sounds are from 20 Hz–70 Hz, although they are much less audible, mainly occurring in children and pathological subjects. Murmurs are usually found in slightly higher frequencies and range from 25 Hz to 400 Hz [10], with some being found in frequencies higher than 600 Hz, but with far less energy [11].

ECG signals represent the heart's electrical activity [12]. An ECG signal consists of the P, QRS complex, and T waves, with a U wave also occasionally present [13]. These waves can contain information to aid in CVD diagnosis. ECG signals are commonly filtered between 0.5 Hz and 40 Hz to remove baseline wander and unwanted noise and interference [14]. For example, in the case of coronary artery disease patients, studies have documented that symptoms such as T-wave inversion, ST-T abnormalities, left ventricular hypertrophy, and premature ventricular contractions can be observed [15].

		Actual	
		Positive	Negative
Classified	Positive	TP	FP
	Negative	FN	TN

Fig. 1. Confusion matrix.

Combining these two signals has produced superior results compared to classification using a single signal [7], suggesting that relevant features for classification exist within both signals. The increase in performance suggests that utilising synchronised PCG and ECG data will help to create more accurate and robust classifiers.

3. Literature survey

3.1. Model robustness

Tran et al. [16] presented a state-of-the-art framework for enhancing model reliability, focusing on robust generalisation. Robust generalisation allows a model to perform well on data outside the training set [16], encompassing in-distribution (ID) and out-of-distribution (OOD) generalisation [16].

ID generalisation pertains to a model's performance on data within the training distribution but outside the training set, addressing underfitting and overfitting issues [16,17]. OOD generalisation, on the other hand, concerns a model's ability to handle data distributions different from the training set, addressing distribution shifts such as subpopulation shifts, covariate shifts, and domain shifts [16,18].

Perturbation resilience is the ability of a model to handle atypical and significantly different data, including corruption, distortion, artifacts, missing data, gaps, spectral masking, extreme noise, and defective inputs, which is critical in clinical settings.

3.1.1. Measuring model robustness

Table 1 shows formulas for traditional binary classification performance measures derived from the confusion matrix in Fig. 1[19–21]. Sensitivity (recall/true positive rate) and specificity (true negative rate) measure correct classifications of positive and negative cases, respectively [19]. Precision (positive predictive value) and negative predictive value measures correctly classified positive and negative cases among classified cases, respectively [19]. Accuracy measures overall correct classifications [19]. Ideally, all these measures are unity, indicating no false predictions.

While having one target metric is ideal, it is impractical as each metric contains different information and no single measure captures all the information from a confusion matrix [20]. Summary metrics can be biased under certain conditions; for instance, accuracy can be misleading for imbalanced datasets. Matthew's correlation coefficient (MCC) is a better single metric for classifier performance than F scores [22].

This work focuses on ID and OOD performance as the metric for model robustness, focusing on balanced accuracy and MCC in addition to accuracy to present an overall indicator of the performance of the classification model.

Table 1
Traditional measures.

Metric	Formula
Sensitivity	$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$
Specificity	$\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$
Precision	$\text{PPV} = \frac{\text{TP}}{\text{TP+FP}}$
Negative Predictive Value	$\text{NPV} = \frac{\text{TN}}{\text{TN+FN}}$
Accuracy	$\text{acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
Balanced Accuracy	$\text{acc}_{\mu} = \frac{\text{TPR+TNR}}{2}$
F1-Positive-Score	$\text{F}_1^+ = \frac{2 \cdot \text{PPV} \cdot \text{TPR}}{\text{PPV+TPR}}$
F1-Negative-Score	$\text{F}_1^- = \frac{2 \cdot \text{NPV} \cdot \text{TNR}}{\text{PNV+TNR}}$
Matthew's Correlation Coefficient	$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}}$

3.1.2. Model robustness and augmentation

Data augmentation creates new data from existing data to increase the training set's size and variety, typically improving model performance. To improve ID generalisation, providing more training data from the same distribution as the original data helps the model generalise to similar examples [16]. To enhance OOD generalisation, extending the training data distribution beyond the original dataset, such as by balancing labels or adding scarce feature combinations, helps the model handle distribution shifts more effectively [23].

3.2. Generative models

Generative models are trained to learn the underlying distribution of the data to generate new samples. As such, the goal is to train a mapping between the latent space and the data space so that the resulting samples are similar to the original data. One of the important properties of the latent space is that it can enable the creation of new data through the manipulation of semantic representations of features and labels. In recent history, three classes of models have advanced the field of generative learning in waves.

These classes are Autoencoders (AEs), Generative Adversarial Networks (GANs) and Diffusion models (DMs). The first class of models, AEs, encode input data to a lower-dimensional latent space and then decode it back to the data space, often used in denoising models due to their ability to reconstruct the input from the latent space [24]. Variational Autoencoders (VAEs), an extension of AEs, regularise the latent distribution, enabling meaningful sampling from the latent space and removing discontinuities, thus facilitating generative capabilities [25]. GANs, the second class, consist of a generator and a discriminator network; the generator creates realistic samples from random noise, while the discriminator attempts to distinguish between real and synthetic samples, engaging in a zero-sum game to improve both networks [26]. DMs, the third class, add random noise to input data and then train the model to reverse this process, learning to denoise data in a structured manner, with models like Latent Diffusion Models (LDMs) performing diffusion in the latent space for computational efficiency [27–29].

The "generative learning trilemma" may guide the trade-offs in choosing a generative learning model. As Fig. 2 (adapted from [30]) shows, models often excel at only two of three desired goals: high sample quality, fast sample speed, and large sample variety. However, as mentioned earlier, performing the diffusion process in latent space allows LDMs to generate samples much faster, such that some argue it bypasses the trilemma in practice [29,30]. For this reason, LDMs have seen recent use in expanding datasets in biomedical projects, where data collection is prohibitively costly [31]. As such, this work aims to use both the WaveGrad and DiffWave diffusion models for the creation of PCG from ECG signals.

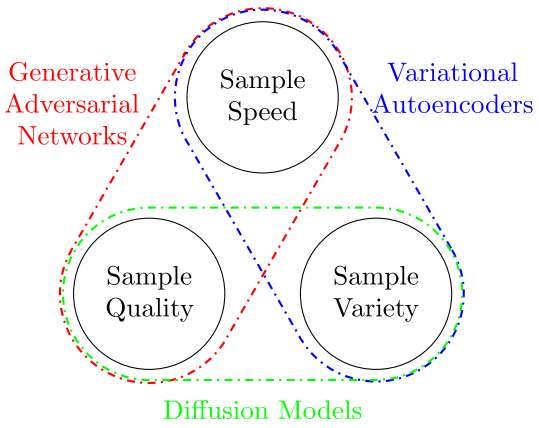


Fig. 2. The generative learning trilemma.

3.3. Biomedical signal augmentation

In [4], data augmentation was employed to expand a PCG dataset from 3153 recordings to 53 601 recordings, an increase by a factor of 17. The augmentation included a random combination of effects such as changes to pitch, speed, tempo, dither, volume, and mixing with audio [4]. Despite achieving a sensitivity of 96 % and a specificity of 83 %, the authors concluded that their approach did not generalise well, with performance varying from 99 % on the dataset with the most recordings to 50 % on the dataset with the fewest recordings [4]. Consequently, Thomae and Dominik [4] suggested that more training data and further augmentation is necessary to enhance performance on unseen data.

In a subsequent study by Zhou et al. [32], models trained with various augmentations were compared against a baseline. Augmentations were applied to both the original and image-transformed data and were categorised by a "physiological constraint" (whether the transform alters or violates physiological possibilities) and/or a "spectrogram constraint" (whether the transform alters the meaning of the spectrogram output) [32]. Augmentations that violated the "spectrogram constraint" were linked to decreased model performance, while adherence to physiological possibilities was associated with improved performance [32]. Notably, no single augmentation improved performance across all metrics, though some offered a more favourable trade-off than others [32].

VAEs have been explored for the generation of synthetic lung auscultation sounds [33], where it was found that the use of VAE-generated signals in the training of classifiers were often improved, but not always, over training on just the original data.

GANs have also found lots of use within biomedical applications [34–36]. The introduction of synthetic data helps to overcome data imbalances as well as improve model performance. In particular, GANs have been used to generate synthetic heart signals [36]. This work found that during early training, the waveform generated resembled a real signal with added noise [36]. Using the Empirical Wavelet Transform (EWT) to reduce this noise, the resulting signal at 2000 epochs was more realistic than the resulting signal at 12 000 epochs, allowing for a sixfold reduction in training time [36]. Further work was performed to show that the generative model had not simply learned the training dataset [36]. As a result, the classifiers were able to classify the synthetic heart sounds correctly with accuracy greater than 90 % [36].

In [37], the general problem of generating synthetic one-dimensional biosignals are explored. Both an autoencoder and GAN-based approach were explored. To evaluate their models, the synthetic and real datasets are each used as either the training or test set for a classifier model that had previously achieved an accuracy

of 99 % [37]. The results from this work showed that the synthetic data captured the underlying features and distributions of the real data and the synthetic data could be used to train classifiers such that they perform well on real data [37]. In addition to this, it was noted that the generative models were readily able to capture the noise of the input data [37].

It was found that although GANs have found lots of use traditionally, the number of papers in medical imaging that utilise VAEs and DMs has increased in recent years. For DMs in particular, there has been a substantial increase in papers, which authors attributed to their ability to generate high-quality images with good mode coverage [38]. Despite the abundance of diffusion models in medical imaging, we could not find, to the best of our knowledge, any use in biomedical audio signals, leaving room for exploration.

3.4. Conditional Denoising Diffusion Probabilistic models

Denoising Diffusion Probabilistic models (DDPM) are a type of diffusion model that follows a Markov process that continuously noises the input, with the network learning to reverse this process by estimating the noise that was added. Further details on the mechanisms of these models are found in Appendix A. Conditional diffusion models for conditional audio generation can be adapted from the diffusion model setup in [39].

3.4.1. WaveGrad

WaveGrad is a DDPM for audio synthesis using conditional generation. The model utilises the architecture consisting of multiple upsampling blocks (UBlocks) and downsampling blocks (DBlocks), with the input signal and the conditioning signal as inputs into the network. The conditioning signal is converted to a mel-spectrogram representation before being input to the model [6]. These UBlocks and DBlocks follow the architecture of the upsampling and downsampling blocks utilised in the Generative Adversarial Network text-to-speech (GAN-TTS) model [40]. The feature-wise linear modulation (FilM) modules combine information from the noisy waveform and the conditioning mel-spectrogram [6]. The UBlock, DBlock and feature-wise linear modulation (FiLM) modules are shown in Fig. 3, with Fig. 4 showing the entire WaveGrad architecture. The loss function is based on the difference between the noise added in each step of the forward diffusion process and the noise predicted during the reverse process [6] as described in Eq. (1), with the Markov process being conditioned on the continuous noise level instead of the time-step. Also, note that the L1 norm was used over the L2 norm as it was found to provide better training stability [6]. WaveGrad only includes a local conditioner in the form of a conditioning signal.

$$\mathbb{E}_{\bar{\alpha},\epsilon}\left[\left\|\epsilon_\theta\left(\sqrt{\bar{\alpha}}\mathbf{y_0}+\sqrt{1-\bar{\alpha}}\epsilon,\mathbf{x},\sqrt{\bar{\alpha}}\right)-\epsilon_t\right\|_1\right] \tag{1}$$

3.4.2. DiffWave

DiffWave is another DDPM for raw audio synthesis with conditional and unconditional generation. The loss function utilises a single ELBO-based training objective without auxiliary losses [5], as described in Eq. (2). One-dimensional convolutions are used on the input and conditioning signals that go through multiple fully connected layers. The model contains a WaveNet [41] *backbone*, consisting of bi-directional dilated convolutions and residual layers and connections. The architecture is shown in Fig. 5. DiffWave can be used for both conditional and unconditional generation. For conditional generation, it uses a local conditioning signal and a global conditioner (discrete labels) [5].

$$\mathbb{E}_{t,\epsilon}\left[\left\|\epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{y_0}+\sqrt{1-\bar{\alpha}_t}\epsilon,\mathbf{x},t\right)-\epsilon_t\right\|_1\right] \tag{2}$$

Table 2
Summary of challenge data.

Database information			Proportion of recordings (%)		
Challenge use	Dataset	Source database	Abnormal	Normal	Unsure
Training	training-a	MITHSDB	67.5	28.4	4.2
	training-b	AADHSDB	14.9	60.2	24.9
	training-c	AUTHHSDB	64.5	22.6	12.9
	training-d	UHAHSDB	47.3	47.3	5.5
	training-e	DLUTHSDB	7.1	86.7	6.2
	training-f	SUAHSDB	27.2	68.4	4.4
		Average	18.1	73.0	8.8
Test	test-b	AADHSDB	15.6	48.8	35.6
	test-c	AUTHHSDB	64.3	28.6	7.1
	test-d	UHAHSDB	45.8	45.8	8.3
	test-e	DLUTHSDB	6.7	86.4	6.9
	test-g	TUTHSDB	18.1	81.9	0.0
	test-i	SSHHSDB	60	34.3	5.7
		Average	12.0	77.1	10.9

4. Materials and methods

To achieve a more robust model, the augmented training dataset must first be created. Fig. 6 depicts the dataset creation process. Once this dataset is created, various classification models can be trained and evaluated to measure the increase in ID and OOD performance.

4.1. Datasets

4.1.1. PhysioNet and computing in cardiology challenge 2016 dataset

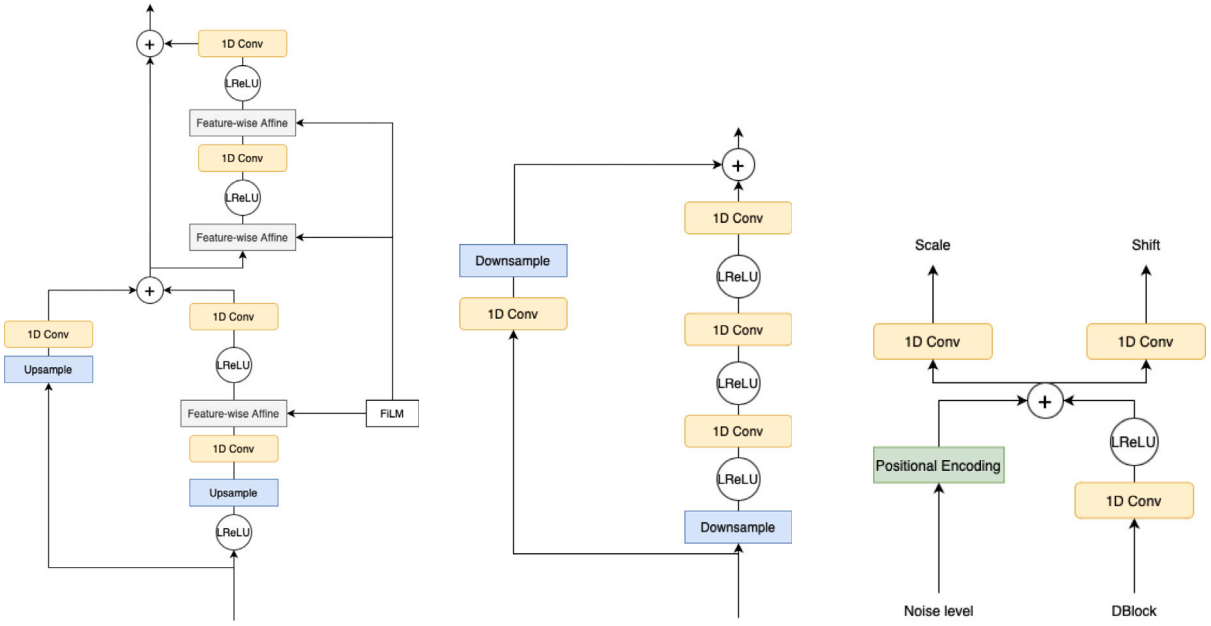
The PhysioNet and Computing in Cardiology Challenge 2016 (CinC) was an international competition that aimed to encourage the development of heart sound classification algorithms [8]. The data was sourced from nine independent databases but excluded a database focused on fetal and maternal heart sounds [8]. Across the nine databases, there are 2435 recordings sourced from 1297 patients [8]. Excluding the aforementioned database and splitting longer recordings into smaller samples, there were in total 4430 samples from 1072 patients, equating to 233 512 heart sounds, 116 865 heart beats, and nearly 30 hours of recordings used in the competition [42]. At the time of (their) publication, this amounted to the largest open-access heart sound database in the world [42].

The recordings were resampled to 2000 Hz for the competition and only one PCG lead was used, with the exception of training-set *a*, which includes ECG [42].

Recordings were divided into either *normal* (healthy), *abnormal* (diagnosed with CVD or other cardiac problems), or *unsure* (low quality signals) [8]. A summary of the data, shown in Table 2, was adapted from [8,42]. These datasets also include additional information, such as individual disease diagnoses and annotations of the heart cycles. These can be used to assist with the data augmentation.

4.1.2. Synchronised multichannel PCG and ECG dataset

Recently, synchronised multichannel PCG and ECG (SMPECG) data has been collected from an EPCG device that consists of seven PCG and one lead-I ECG sensors [43]. Using this device, data was collected from 105 subjects, of which 46 were diagnosed with coronary artery disease. Ten seconds of audio were recorded for each subject, during which the subjects were instructed not to breathe to eliminate lung sounds from the recording. This data was collected in a clinical environment with background noise and non-optimal sensor placement as it is designed for ease of use, making it a challenging dataset for classification, which is representative of a real-world dataset. As only single channel PCG is available in the other datasets, only a single channel (channel 2) was used for this dataset.



(a) UBlock Module Architecture. (b) DBlock Module Architecture. (c) FiLM Module Architecture.

Fig. 3. WaveGrad module architectures.

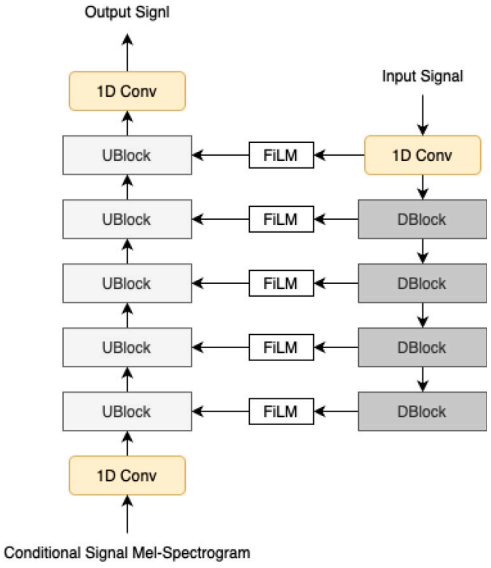


Fig. 4. WaveGrad architecture.

4.1.3. Incentia dataset

Along with the training-a dataset used for the inputs for training the generative models, the incentive dataset [44] was utilised to provide unique unseen ECG to generate an accompanying PCG signal. This data set contains 11,000 patients and 2,774,054,987 labelled heartbeats at a sample rate of 250 Hz with 541,794 segments. Each beat was classified with a type from normal, premature atrial contraction, premature ventricular contraction and rhythm from normal sinusal rhythm, atrial fibrillation and atrial flutter.

4.1.4. Further datasets

To improve the model's robustness against noise, one of the stages of augmentation introduces noise from other PCG and ECG datasets. These are the electro-phono-cardiogram (EPHNOGRAM) dataset [45] for PCG and the Massachusetts Institute of Technology - Beth Israel Hospital (MIT-BIH) dataset [46] for ECG. The EPHNOGRAM dataset comprises 24 healthy adults and contains recordings taken during stress tests and at rest [45]. The MIT-BIH dataset contains 12 half-hour ECG recordings and three half-hour recordings of noise typical in ambulatory ECG recordings, where this noise is used for augmentation [46].

4.2. Signal augmentation

The augmentation procedure of the PCG and ECG signals is shown in Fig. 7. The time stretching augmentation is synchronised to ensure that they are both stretched the same amount, with the black lines representing the flow of the ECG data and the white lines representing the flow of PCG data. Augmentation stages have different percentage chances of occurring, where the chances chosen were determined to provide the widest variety of augmented signals after every stage has been completed whilst also resulting in the best performance. The augmentations vary slightly between PCG and ECG to best meet the physiological constraints.

The PCG signals are augmented in various ways: harmonic percussive source separation (HPSS) for emphasis on certain parts of the signal, time stretching, emphasis on certain bands of the signal using a parametric equalisation (EQ) filter and introducing noise from the EPHNOGRAM dataset [45]. Before these operations are applied, the signals are normalised to have a zero mean and be between −1 and 1. Shown in Fig. 7 is the augmentation procedure applied to PCG data, noted with the white lines.

The HPSS has a 75% chance of occurring and works by extracting harmonic and percussive components of the signal with varying thresholds to extract different parts of the signal. The HPSS implementation is from the librosa v0.1.0 Python library [47,48]. $\mathbf{X}(t,k)$ denotes the short-time Fourier transform (STFT) of the signal $\mathbf{x}(t)$, defined as

$$\mathbf{X}(t,k) = \sum_{n=0}^{N-1} \mathbf{w}(n)\mathbf{x}(n+tH)\exp(-2\pi jkn/N) \tag{3}$$

where $\mathbf{w}$ is a sine-window, H represents the hop size and N is the window length and the length of the discrete Fourier transform.

Firstly, the STFT of the signal is calculated, with the parameters chosen randomly from a window length of 512, 1024 and 2048 with

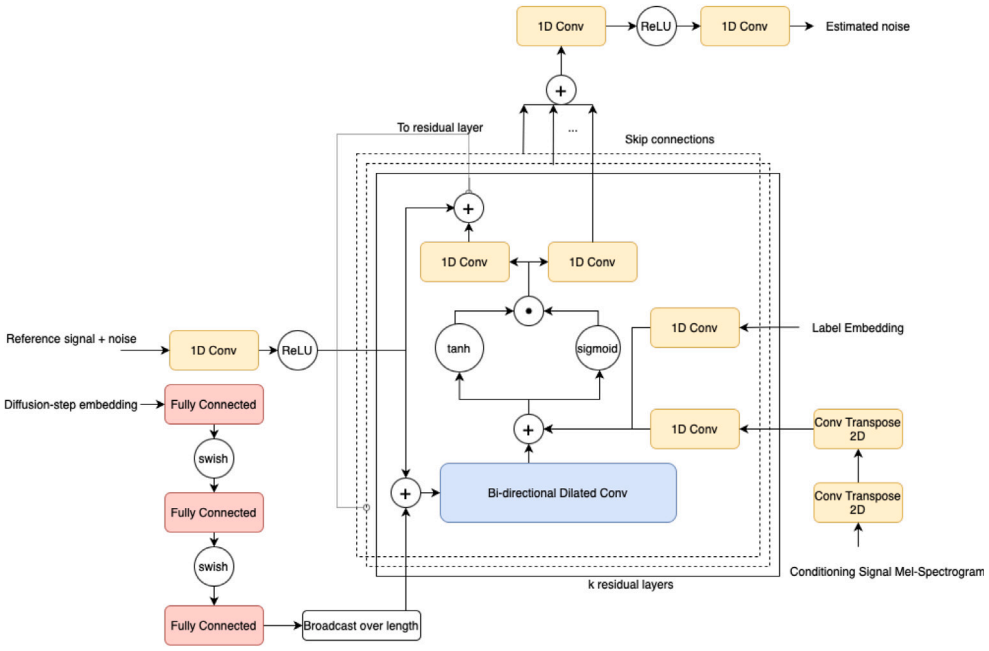


Fig. 5. DiffWave architecture.

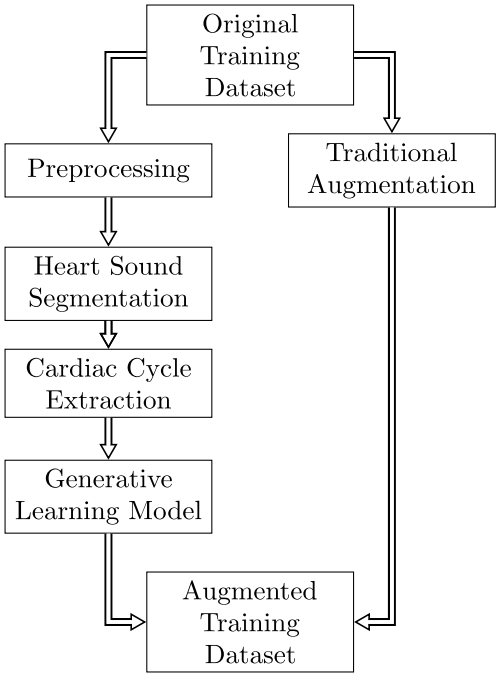


Fig. 6. Data augmentation architecture.

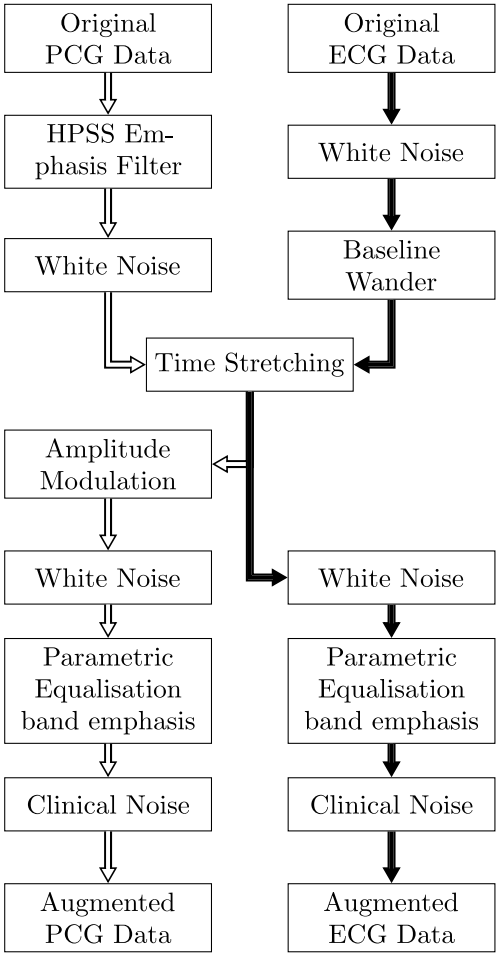


Fig. 7. PCG and ECG traditional augmentation procedure.

equal probability. A hop length was randomly chosen from 16, 32, 64, and 128 with uniform distribution.

Following this, the harmonic and percussive components are then extracted from the following,

$$\tilde{\mathbf{Y}}_h(t,k) = median(\mathbf{X}(t-\ell_h,k),\ldots,\mathbf{X}(t+\ell_h,k)) \tag{4}$$

$$\tilde{\mathbf{Y}}_p(t,k) = median(\mathbf{X}(t,k-\ell_p),\ldots,\mathbf{X}(t,k+\ell_p)) \tag{5}$$

$$\mathbf{M}_h(t,k) = \begin{cases} 1, & \text{if } \frac{\tilde{\mathbf{Y}}_h(t,k)}{\tilde{\mathbf{Y}}_p(t,k)+\eta} > \lambda_h \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$\mathbf{M}_p(t,k) = \begin{cases} 1, & \text{if } \frac{\tilde{\mathbf{Y}}_p(t,k)}{\tilde{\mathbf{Y}}_h(t,k)+\eta} \geq \lambda_p \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$\mathbf{X}_h(t,k) = \mathbf{X}(t,k) \cdot \mathbf{M}_h(t,k) \tag{8}$$

$$\mathbf{X}_p(t,k) = \mathbf{X}(t,k) \cdot \mathbf{M}_p(t,k) \tag{9}$$

where $\mathbf{X}_h(t,k)$ is the harmonic component, $\mathbf{X}_p(t,k)$ is the percussive component η is a small number added to avoid a divide by 0 error [48]. $\mathbf{x}_h(t)$ and $\mathbf{x}_p(t)$ are the inverse STFT (ISTFT) of $\mathbf{X}_h(t,k)$ and $\mathbf{X}_p(t,k)$. If the thresholds, $\lambda_h > 1$ or $\lambda_p > 1$, there will be some part of the spectrum that is not a harmonic or percussive component of the signal but a residual component that appears as textured noise. As the abnormalities to be detected are from diseases that produce more percussive or harmonic sounds, these residuals can be ignored without important information loss that would negatively impact the ability of a classifier to classify these sounds.

The first set have parameters $\lambda_h = rand(1,2)$, $\lambda_p = rand(1,2)$, $\ell_h = randint(5,30)$, and $\ell_p = randint(5,30)$. *rand* denotes a random floating point number chosen uniformly between the two bounds, and *randint* is an integer uniformly chosen between those bounds. The second set are then extracted from $\mathbf{X}_h(t,k)$ and $\mathbf{X}_p(t,k)$. $\mathbf{X}_{hh}(t,k)$ and $\mathbf{X}_{hp}(t,k)$ are the harmonic and percussive components of $\mathbf{X}_h(t,k)$ and $\mathbf{X}_{ph}(t,k)$ and $\mathbf{X}_{pp}(t,k)$ the harmonic and percussive components of $\mathbf{X}_p(t,k)$. The second stage of decomposition uses parameters of $\lambda_{hh} = rand(1,4)$, $\lambda_{hp} = rand(1,4)$, $\lambda_{ph} = rand(1,4)$, $\lambda_{pp} = rand(1,4)$, $\ell_{hh} = randint(5,30)$, $\ell_{hp} = randint(5,30)$, and $\ell_{ph} = randint(5,30)$, $\ell_{pp} = randint(5,30)$.

The ISTFT is then applied to each component before reconstructing the signal as,

$$\mathbf{s}_{HPSS}(t) = a_{hh}\mathbf{x}_{hh}(t) + a_{hp}\mathbf{x}_{hp}(t) + a_{ph}\mathbf{x}_{ph}(t) + a_{pp}\mathbf{x}_{pp}(t) \tag{10}$$

where $a_{hh} = rand(0.01,10)$, $a_{hp} = rand(0.01,10)$, $a_{ph} = rand(0.01,10)$, $a_{pp} = rand(0.01,10)$.

This two stage decomposition and reconstruction described in Eq. (10) is done twice to create $\mathbf{s}_{HPSS_1}(t)$ and $\mathbf{s}_{HPSS_2}(t)$, which are then combined to get the final augmented signal $\mathbf{s}_{HPSS_{final}}(t)$,

$$\mathbf{s}_{HPSS_{final}}(t) = \mathbf{s}_{HPSS_1}(t) + a_{HPSS}\mathbf{s}_{HPSS_2}(t) \tag{11}$$

where $a_{HPSS} = rand(0.01,0.05)$. The use of these parameters was determined by inspection to ensure the signals remain realistic.

Next, there is a 7.5% chance of introducing noise to the signal, as defined in the equation below, where $\mathbf{s}_{HPSS}(t)$ is the signal after the HPSS augmentation stage, $\mathbf{s}_{SN}(t)$ is the augmented signal and $\mathbf{r}(t) \sim \mathcal{N}(\mu, \sigma I)$, $\sigma = rand\_choice(0.01, 0.001, 0.0001)$ and $\mu = rand(0,0.1)$. Note that $\mathbf{s}_{HPSS}(t)$ may not have had the HPSS augmentation applied as it depends on the random chance. *rand_choice*() denotes a random choice from those numbers with equal probability.

$$\mathbf{s}_{SN}(t) = \mathbf{s}_{HPSS}(t) + \mathbf{r}(t) \tag{12}$$

Following this, there is a 25% chance of adding in a time warp. This time warp will stretch the signal randomly to either 1.004 times the length or 1.006 times the length of the original signal. It is noted that a time warp with the same factor will be applied to both the PCG and ECG.

There is then a 75% chance of adding in amplitude modulation. The modulation is done as described in Eq. (13), where $b_{AM_1} = rand(0.01,0.25)$, $b_{AM_2} = rand(0.01,0.25)$, $c_{AM_1} = rand(0.05,0.5)$, $c_{AM_2} = rand(0.001,0.05)$, $d_{AM_1} = rand(0,1)$, $d_{AM_2} = rand(0,1)$ and $s_{TS}(t)$ is signal after the time stretch augmentation stage, which depending on the random chance may have been time-stretched.

$$\mathbf{s}_{AM} = \mathbf{s}_{TS}(t) \cdot \left(1 + b_{AM_1} \sin\left(2\pi c_{AM_1} t + d_{AM_1}\right) + b_{AM_2} \sin\left(2\pi c_{AM_2} t + d_{AM_2}\right)\right) \tag{13}$$

Next, there is another 7.5% chance of introducing the same noise as done in Eq. (12). Following this, there is a 25% chance of applying parametric equalisation to boost frequency bands. Given the frequency range of 2 Hz–500 Hz, the bandwidth is randomly selected between 5% and 20% of this range, and the signal is attenuated using a bandpass filter. After repeating this process 5 times, the filtered signal and original signal are summed and normalised.

Lastly, real noise from the EPHNOGRAM dataset is introduced. The introduced noise from the EPHNOGRAM is clinical noise extracted from some of the recordings in this dataset. This augmentation occurs 50% of the time.

The ECG signals are also augmented in numerous ways; these include introducing random noise, adding baseline wander, time stretching, adding noise from the MIT-BIH dataset, and emphasising certain signal bands. Fig. 7 shows the order of processing on the ECG, indicated with the black lines.

Random noise is applied the same way as the PCG noise, as defined in Eq. (12), with this augmentation occurring with a probability of 7.5%. Next, a baseline wander is added 30% of this time. This is done as described in Eq. (14), where $b_{BW_1} = rand(0.01,0.2)$, $b_{BW_2} = rand(0.01,0.2)$, $c_{BW_1} = rand(0.05,0.5)$, $c_{BW_2} = rand(0.001,0.05)$, $d_{BW_1} = rand(0,1)$, $d_{BW_2} = rand(0,1)$. $\mathbf{s}_{SN_E}(t)$ is the ECG signal after the random noise augmentation stage, which may include the random noise as per the random chance.

$$\mathbf{s}_{BW}(t) = \mathbf{s}_{SN_E}(t) + b_{BW_1} \sin\left(2\pi c_{BW_1} t + d_{BW_1}\right) + b_{BW_2} \sin\left(2\pi c_{BW_2} t + d_{BW_2}\right) \tag{14}$$

Following this, there is a 25% chance of a timewarp between 1 and 1.06 times the original signal. It is noted that a timewarp with the same factor will be applied to both the PCG and ECG. Then, the same parametric equalisation, as with the PCG, is applied between 0.25 Hz and 100 Hz.

Lastly, noise from the MIT-BIH database is added, occuring 50% of the time. This is noise from the ECG sensors taken from recordings in the MIT-BIH database.

4.3. Synthetic audio generation

Synthetic signals were generated using the mel-spectrogram of the ECG signal as a conditioner for both the WaveGrad [6] and DiffWave [5] diffusion models. They are trained before data is generated for use. These diffusion models generated data for 3200 patients, 800 abnormal and 2400 normal, with three segments used to train the classification models. This is done to reduce the effect of overfitting to the synthetic signals. The ECG signals for conditioning were taken from the icentia database [44] to introduce new data, with abnormal ECG used for abnormal PCG. The generative models were trained to create individual conditions and make them more realistic using additional labels from the dataset. To get around the lack of training data, the order of heart cycles was rearranged to increase training diversity. DiffWave and WaveGrad models were trained on an Nvidia RTX 4090 for 24 h. The parameters for the DiffWave model that differ from the default are shown below in Table 3. Parameters used for the WaveGrad model that differ from the default are shown in Table Table 4. Both models differ slightly from their base implementations as they use a custom global conditioner. Additional global conditioners were added for specific abnormalities or lack of abnormalities, such as mitral valve prolapse, innocent or benign murmurs, aortic disease, miscellaneous conditions, and normal.

To ensure a diversity of training examples, various heart cycles were occasionally rearranged for each patient for each minibatch during

Table 3
DiffWave parameters.

Parameter	Value
Residual layers	30
Residual channels	64
Dilation cycle length	10
Embedding dimension	32
Batch size	8
Learning rate	2e−4
Noise schedule	T=50, linearly spaced [1e−4, 5e−2]
Inference noise schedule	{1e−4, 1e−3, 1e−2, 5e−2, 2e−1, 5e−1}

Table 4
WaveGrad parameters.

Parameter	Value
Embedding dimension	32
Batch size	8
Learning rate	2e−4
Noise schedule	T=1000, linearly spaced [1e−6, 1e−2]

Table 5
Adam optimiser parameters.

Parameter	Value
initial learning rate	0.001
betas	(0.9, 0.999)
epsilon	10^{-8}
weight decay	10^{-3}
learning rate step size up	2
learning rate step size down	2
max learning rate	10^{-3}

training. This was done inside a custom collator, with a 75% chance of rearranging the heart cycles. Heart cycles could be rearranged in three ways with equal probability. The first will take groupings of many cycles and then randomly rearrange these large groups. These first groups would have a size of half of the total number of heart cycles within that signal. Secondly, groupings of 1 to 4 heart cycles were chosen randomly and used to rearrange the signal. Finally, the third way involved rearranging each heart cycle.

Although this rearranging can violate physiological constraints, it was found that this helped the model learn a better representation of the data and improved classification results when trained on the synthetic data.

The signals were then bandpass filtered between 2 Hz to 500 Hz for PCG and 0.25 Hz to 100 Hz for ECG, the conditioning signal. A mel-spectrogram of the ECG was created as the local conditioning signal. The mel-spectrogram was created using a sample rate of 4 kHz, window length 1024, hop length 256, and 80 mel bins. Crossfading was used to ensure minimal audio artifacts when rearranging heart cycles. As the signals are joined when they are both in the same state, the end of the cycle in the diastole phase, they are assumed to be roughly correlated. The crossfade occurs between the last 40 samples of the first signal, $-1 \leq t \leq 0$, and the first 40 samples from the second signal, $0 \leq t \leq 1$. If one of the signals has a low variance, then a simple linear crossfade is used between the two. A linear crossfade can be described from Eq. (15) and (16) below,

$$\mathbf{f}(t) = 1/2 + t/2, \quad -1 < t < 1 \tag{15}$$

$$\mathbf{v}(t) = \mathbf{f}(t)\mathbf{y}(t) + \mathbf{f}(-t)\mathbf{x}(t) \tag{16}$$

where f is the crossfade function, v is the final spliced signal, x is the last 40 samples from the first signal, and y is the first 40 samples from the second signal.

Otherwise, the following crossfade function will be used to ensure a crossfade is applied that represents how correlated the two signals are. For two fully uncorrelated signals, a constant power crossfade would be desired, and for two fully correlated signals, a constant voltage crossfade would be desired and something in between if not fully correlated or uncorrelated. Assuming that the crossfade function is deterministic, the two signals are a random process. Along with the assumption, the mean power of the signals at the point of crossfading is equal as they are being crossfaded when in the same phase of the heart cycle. This allows the following generalised crossfade function [49] to be used to satisfy a crossfade related to the signals' correlation. The crossfade is defined in Eq. (17)–(19),

$$\mathbf{o}(t) = \frac{9}{16}\sin\left(\frac{\pi}{2}t\right) + \frac{1}{16}\sin\left(\frac{3\pi}{2}t\right), \quad -1 < t < 1 \tag{17}$$

$$\mathbf{e}(t) = \sqrt{\frac{1}{2(1+r)} - \left(\frac{1-r}{1+r}\right)\mathbf{o}(t)^2} \tag{18}$$

$$\mathbf{f}(t) = \mathbf{o}(t) + \mathbf{e}(t) \tag{19}$$

where e is the even component of the crossfade function, and o is the odd component, and r is the correlation coefficient of the two signals at zero lag and $0 \leq r \leq 1$. The crossfade is then interpolated to double the length using a univariate spline, with a degree of 3 and a smoothing factor equal to the length of the signal. The implementation is the scipy implementation of the univariate spline [50]. The final signal consists of the first signal before the last 40 samples, the crossfaded and interpolated signal, and the second signal after the first 40 samples. Fig. 8 demonstrates the effect that this crossfade has on reducing artifacts. Rearranging of the heart cycles can be seen through the rearranging of the chirp in the last row. The first column shows the original signal, the second shows the rearranging of all heart cycles, the third shows the rearranging of a few heart cycles, and the final shows the rearranging of larger groups of heart cycles.

4.4. Classification model

The model used to test the augmented dataset is a convolutional neural network-based model finetuned from ResNet trained on ImageNet [7]. The purpose of choosing this model is not to show its better performance in classification but to demonstrate the capability of the proposed data augmentation methods. Before the signals are passed into the convolutional neural network (CNN), the PCG signal is bandpass filtered between 45 Hz and 400 Hz. The ECG signal is bandpass filtered between 25 Hz and 100 Hz. The signals also then undergo normalisation. A spectrogram is created from the signal before being passed to the model, with a window length of 100 and a hop length of 50. This spectrogram is created based on 1.5 s of audio, with each being referred to as a fragment, with the training objective to maximise accuracy on the fragment level. From the synthetic data, only three fragments of 1.5 s audio are taken to ensure reduced overfitting to the synthetic data. These 1.5 s fragments differ from the original model [7] which took in a single heart cycle. This change has been done to reduce the need for accurate segmentation. For testing the subject level, the outputs from the classification are averaged between all fragments before the classification is made, as was done previously. The Adam optimiser is used for training along with a cyclic triangular learning rate scheduler with parameters below in Table 5.

During the model's training on the original dataset, as a CNN is being finetuned, only 10 epochs are used in which the best weights are chosen from the highest MCC value from the validation set to reduce overfitting. The model is only updated for each dataset if it performed better on the validation set than previously. A schedule is used to reduce the overfitting of the synthetic data for training on the augmented dataset. This schedule can be found below in Table 6 and was experimentally determined to provide the best results, where

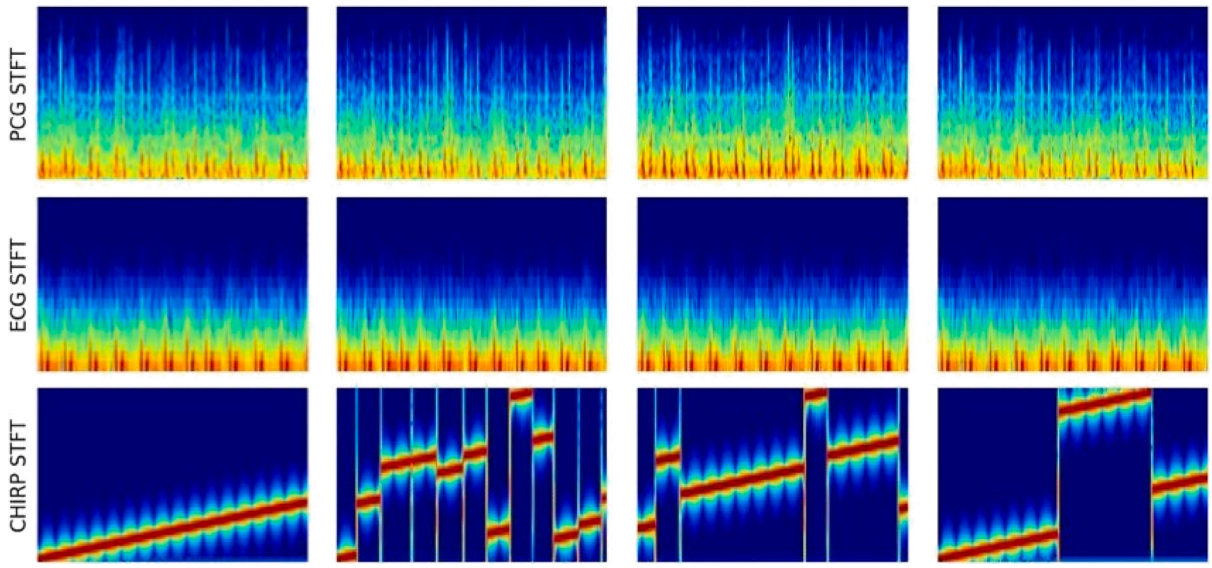


Fig. 8. Rearranged heart cycles with crossfade.

Table 6
Training schedule.

Data	Epochs
max-mix	8
max-aug	8
max-mix	8
max-aug	8
max-mix	8
max-aug	8
max-mix	16
max-aug	16
max-mix	16
max-aug	16
max-mix	16
max-aug	16

max-mix is all of the data with no augmentations being applied to the original dataset and 3 augmentations applied to the DiffWave and WaveGrad data. From the synthetic data, only three random segments were taken to ensure the model does not overfit to the synthetic data. The max-aug data is the original data with 30 augmentations being applied and no synthetic data.

As only the training-a dataset contains synchronised PCG and ECG for measuring the OOD performance, a PCG-only model will also be trained and used to be evaluated on training-b-f datasets whilst the PCG and ECG input model will be evaluated on the SMPECG dataset.

5. Results

These results are to demonstrate the performance improvement observed in deep learning models when training is conducted on the augmented dataset. We do not aim to evaluate the performance of the convolutional neural network.

5.1. In-distribution performance

The ID results are for the datasets on which the models were trained. This shows the increase in performance when training on the augmented dataset compared to the original dataset. As the only dataset being trained on was training-a, these are the only models presented for in-distribution performance. Table 7 displays the ID performance when the models are trained on the original dataset, with Table 8 displaying the ID performance for models trained on the augmented dataset.

5.2. Out-of-distribution performance

The out-of-distribution results are for the datasets the models were not trained on. Hence, this shows an increase in the generalisation of the models to other datasets that were not trained on. As the dataset being trained on was training-a, all other datasets are presented for the out-of-distribution performance. Table 9 shows the OOD performance on the original dataset, with Table 10 showing the OOD performance when trained on the augmented dataset.

5.3. Synthetic generation output

Fig. 9 shows the generation output of the DiffWave diffusion model, with the other diffusion models, WaveGrad, output in Fig. 10. The reference signal is from patient ‘a0040‘ from the training-a dataset. It is noted that this generation is done from random noise and the conditioning signal, so it will not be reconstructed to look identical to the reference signal. However, it should be time-aligned with the conditioning signal. To better demonstrate the improved generation of the proposed diffusion model, Fig. 11 and Fig. 12 show the outputs from a conditional β variational autoencoder (cβ-VAE) and a conditional deep convolutional generative adversarial network (cDCGAN). Appendices B and C contain the architectures and training used for the cβ-VAE and cDCGAN models, respectively. It was found that the cDCGAN could not be trained to generate realistic PCG signals as it suffered from mode collapse and struggled with generation. The cβ-VAE resulted in a noisier generated PCG signal than the diffusion model. For this reason, only diffusion models were utilised for the synthetic data generation. These results are inline with the generative trilemma as mentioned in Section 3.2. Further synthetic signals of the diffusion models can be found in Appendix D.

6. Discussion

It was found that the ID performance was improved for all models tested, with a 2.5% improvement in accuracy for the PECG model and a 13.6% improvement in subject-level accuracy for the PCG model. The augmented dataset is also shown to improve the balanced accuracy and hence help to balance between sensitivity and specificity, with all these being improved from the original dataset to the augmented dataset. This was observed through a balanced accuracy improvement of 4.1% and 24.2% for the PECG and PCG models, respectively. This is further

Table 7
Models performance ID trained on the original dataset.

Dataset	Data	Acc	Acc-mu	TPR	TNR	PPV	NPV	$F1^+$	$F1^-$	MCC
training-a	PCG+ECG	90.10%	89.40%	91.20%	87.50%	94.50%	80.80%	92.90%	84.0%	0.770
training-a	PCG	70.40%	56.00%	91.20%	20.80%	73.20%	50.00%	81.20%	29.40%	0.167

Table 8
Models performance ID trained on the augmented dataset.

Dataset	Data	Acc	Acc-mu	TPR	TNR	PPV	NPV	$F1^+$	$F1^-$	MCC
training-a	PCG+ECG	92.60%	93.50%	91.20%	95.80%	98.10%	82.10%	94.50%	88.50%	0.836
training-a	PCG	84.00%	80.20%	89.50%	70.80%	87.90%	73.90%	88.70%	72.30%	0.611

Table 9
Models performance in OOD trained on the original dataset.

Dataset	Data	Acc	Acc-mu	TPR	TNR	PPV	NPV	$F1^+$	$F1^-$	MCC
training-b	PCG	22.90%	50.7%	99.00%	2.30%	21.5%	90.00%	35.30%	4.5%	0.040
training-c	PCG	74.20%	47.90%	95.80%	0.00%	76.70%	0.00%	85.20%	NaN	−0.099
training-d	PCG	49.10%	48.50%	82.10%	14.80%	50.0%	44.40%	62.20%	22.20%	−0.041
training-e	PCG	40.90%	65.80%	96.20%	35.50%	12.70%	99.00%	22.50%	52.20%	0.192
training-f	PCG	52.60%	58.60%	73.50%	43.80%	35.70%	79.50%	48.10%	56.50%	0.162
SMPECG	PCG+ECG	56.20%	50.20%	98.30%	2.20%	56.30%	50.00%	71.60%	4.20%	0.017
SMPECG	PCG	56.20%	50.20%	98.30%	2.20%	56.30%	50.00	71.60%	4.20%	0.017

Table 10
Models performance in OOD trained on the augmented dataset.

Dataset	Data	Acc	Acc-mu	TPR	TNR	PPV	NPV	$F1^+$	$F1^-$	MCC
training-b	PCG	33.30%	53.10%	87.50%	18.70%	22.50%	84.70%	35.80%	30.60%	0.066
training-c	PCG	83.90%	74.70%	91.7%	57.10%	88.00%	66.70%	89.80%	61.50%	0.517
training-d	PCG	52.70%	52.00%	92.90%	11.10%	52.00%	60.00%	66.70%	18.80%	0.069
training-e	PCG	84.00%	86.00%	88.50%	83.50%	34.50%	98.70%	49.60%	90.50%	0.489
training-f	PCG	73.70%	60.10%	26.50%	93.80%	64.30%	75.00%	37.50%	83.30%	0.282
SMPECG	PCG+ECG	61.90%	57.00%	96.60%	17.40%	60.00%	80.00%	71.40%	28.60%	0.237
SMPECG	PCG	57.10%	51.60%	96.60%	6.50%	57.00%	60.00%	71.70%	11.80%	0.073

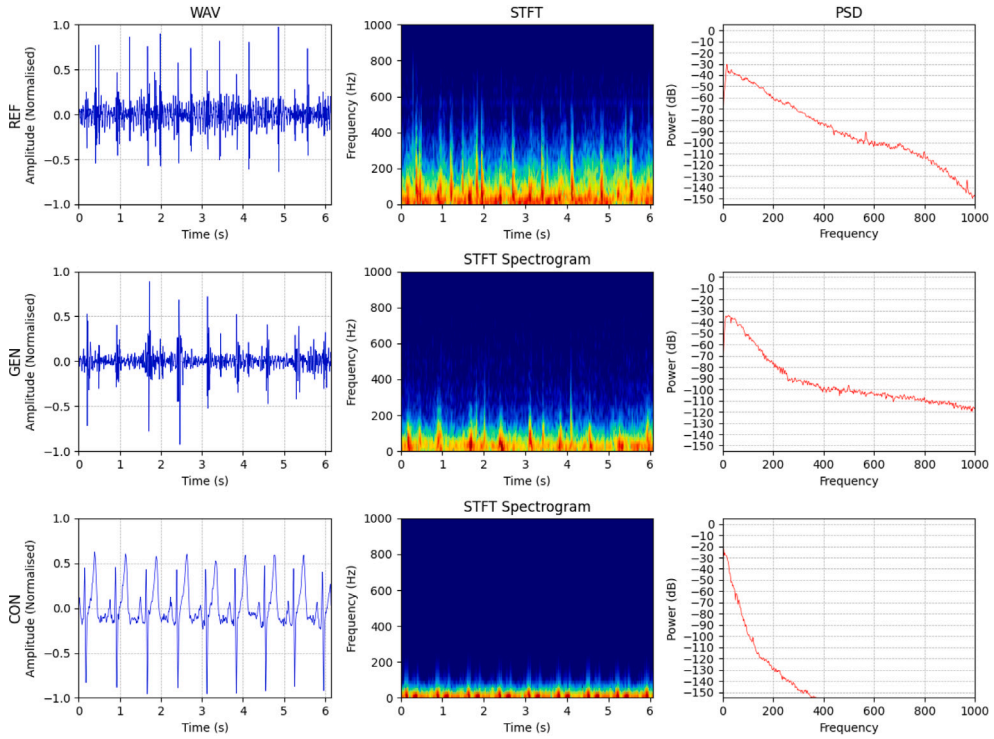


Fig. 9. Generated DiffWave signal.

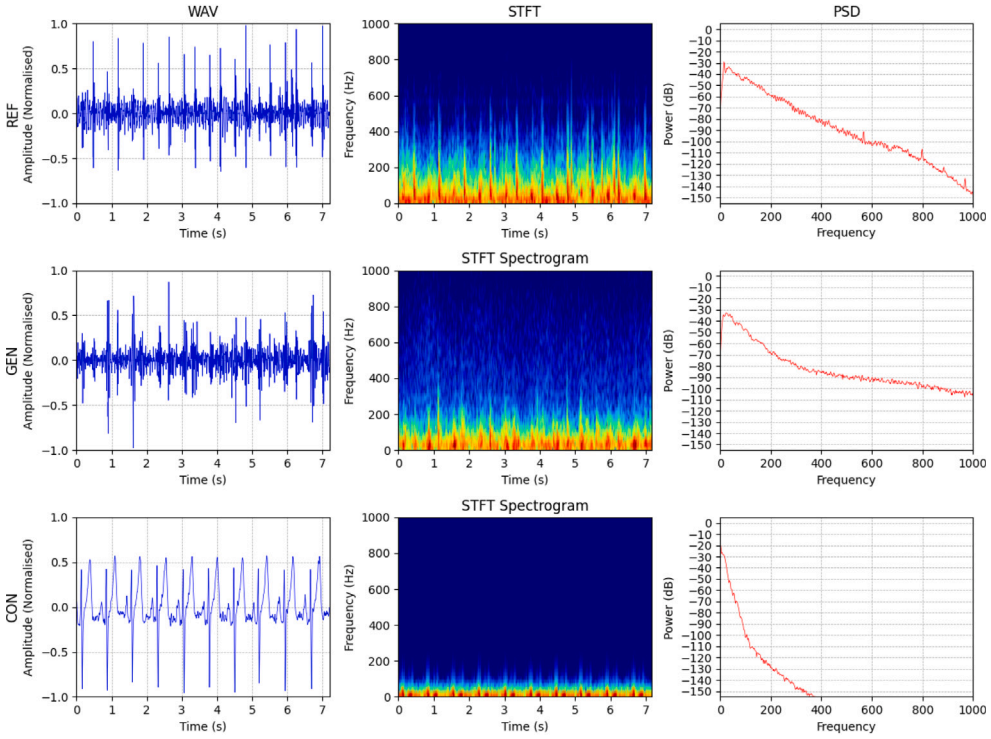


Fig. 10. Generated WaveGrad signal.

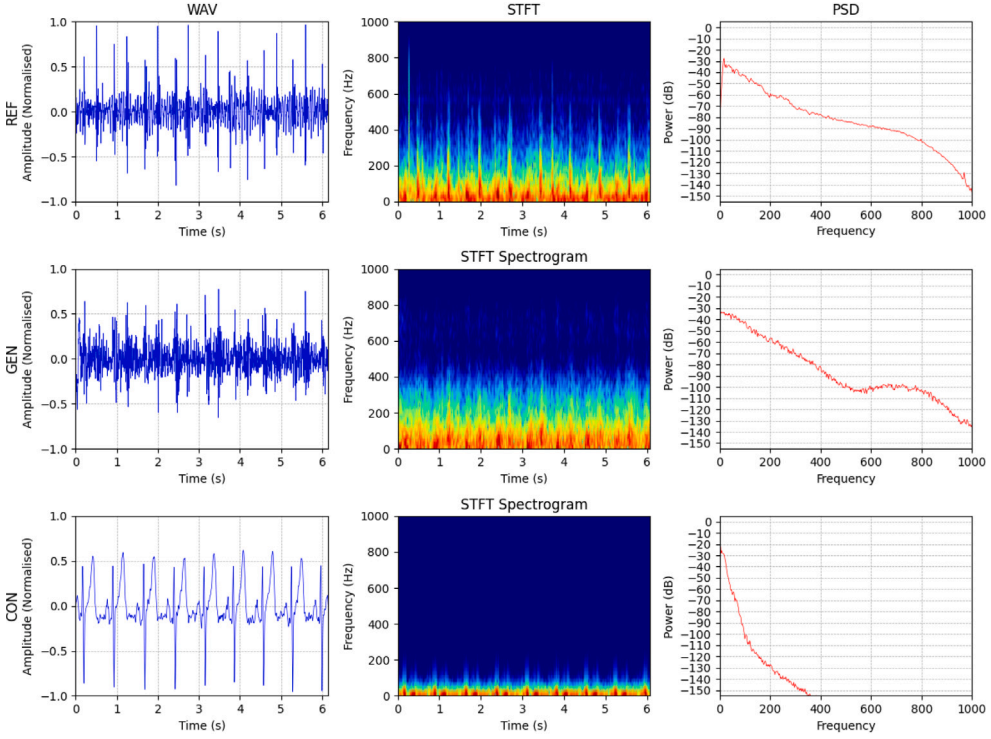


Fig. 11. Generated cβ-VAE signal.

shown by an increase in the MCC value from 0.77 to 0.836 and 0.167 to 0.611 for the EPCG and PCG models, respectively. This shows that by augmenting the original data as well as adding synthetic data, and ensuring a balanced dataset, the ID performance can be improved.

The OOD performance was also seen to improve with the augmented dataset. Although the models were not trained on these datasets, the introduction of augmented data improved all model's accuracy and overall robustness, as seen by the increase in MCC values across all

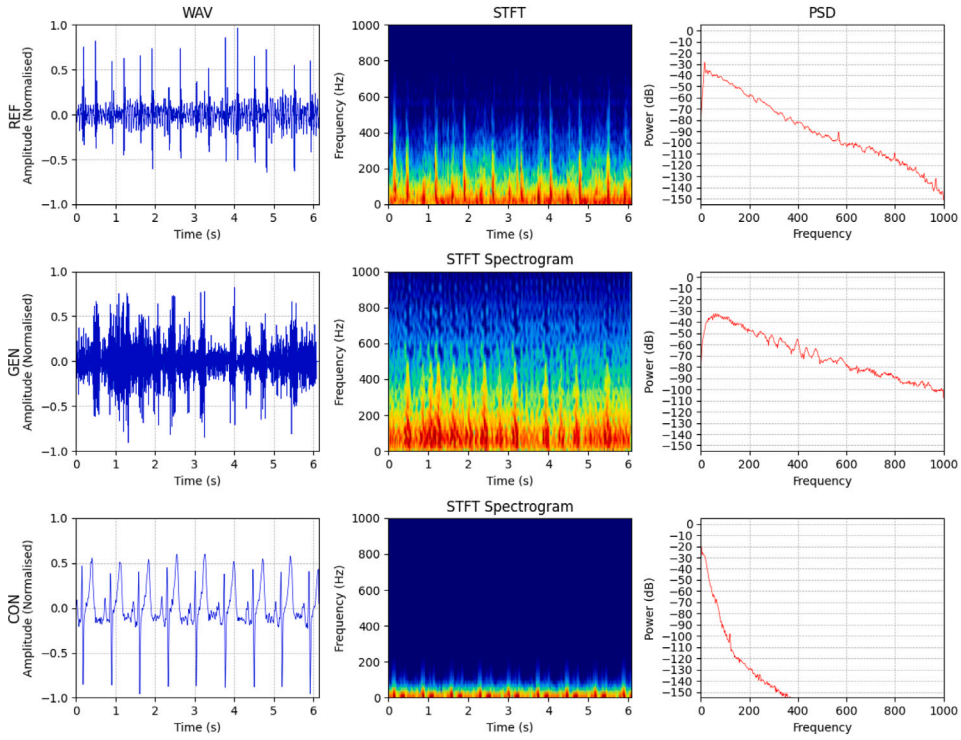


Fig. 12. Generated cDCGAN signal.

datasets. In particular, in the CinC datasets, there was an improvement in accuracy of at most 43.1% in training-e and of at least 3.6% in training-d, with the improvement in accuracy in all other CinC datasets are between these values. Further, the balanced accuracy in all of these datasets was improved. With the greatest increase in balanced accuracy of 26.8% from training-c and the smallest being 1.5% from training-f. The MCC was also seen to increase in all cases, with the greatest increase of 0.616 occurring in training-c and the smallest increase of 0.026 in training-b. With all performance metrics increasing, the OOD performance was improved by the use of this augmented dataset, which shows that these augmentations help to improve the robustness of models when used on unseen OOD data.

In the SMPECG dataset, there was a much smaller improvement in accuracy, with an increase of 5.7% with the EPCG model and an increase of 0.9% with the PCG model. Also, balanced accuracy for both models increases, with 6.8% and 1.4% for the EPCG and PCG models, respectively. However, there was a much greater improvement in MCC and overall balancing the performance with an increase to the MCC value of 0.22 for the EPCG model and 0.056 for the PCG model. This shows that although a small improvement, this augmentation helps not only improve classification accuracy but also helps to balance the classifier, improving its balanced accuracy and MCC values.

As shown, both the ID and OOD performance have been increased by utilising the augmented data, achieving the objective of improving the robustness of the classifier. Better results are found for PCG-only models. This, however, is due to more data to test with than synchronised PCG and ECG data. However, the OOD for some datasets is still low, showing that there is still room for improvement in making a truly robust and general abnormal heart sound classifier. Utilising a larger dataset and applying these methods, the classifier is expected to become much more general, as seen with classifiers trained on this smaller dataset. The training datasets and the CinC testing datasets are also not as representative of real-world datasets as compared to the SMPECG dataset; however, with the improvement seen on this dataset, it suggests that when used entirely on real-world datasets, this method will still result in improvement.

7. Conclusion and further work

Increasing training data through augmentation has improved ID and OOD performance in classifying abnormal heart sounds. The use of diffusion models to generate synthetic heart sounds conditioned on ECG signals has successfully enabled the generation of synchronised PCG from ECG data, expanding the data distribution and enhancing classifier robustness. This is not limited to classifiers that utilise multimodal PCG and ECG data but also for single-mode classifiers that utilise only PCG, as found from the increase in performance and robustness of PCG-only models. Future work should scale this approach to multichannel PCG signals for use with classifiers that utilise such data.

This study provides evidence that data augmentation, specifically through DDPMs, can significantly enhance the robustness and generalisation of classifiers for abnormal heart sound detection. By conditioning synthetic PCG signals on ECG data, we generated augmented datasets that improved performance in both ID and OOD scenarios, consistently observed across key metrics such as accuracy, balanced accuracy, and MCC.

Our approach increases the size of training datasets and enriches data diversity, which is crucial for developing models resilient to variations in real-world clinical settings. The augmentation process effectively addresses data imbalance and noise, providing a stronger foundation for training machine learning models.

However, while the introduced augmentation techniques have shown promise, certain limitations remain, particularly in generalising models to new datasets. The OOD performance, though improved, suggests that further refinement of these methods is necessary. This could involve optimising diffusion model parameters or exploring alternative generative approaches that better capture the complex patterns in biomedical signals.

Future work should focus on scaling these methods to accommodate multichannel PCG data, enabling more comprehensive heart sound analysis and potentially improving classification accuracy. This will allow the training and test datasets to use data from the SMPECG

dataset, demonstrating the effectiveness of this methodology on a real-world dataset. This study demonstrates a viable strategy for enhancing classifier performance through synthetic data generation, contributing to more reliable cardiovascular disease diagnosis.

CRediT authorship contribution statement

Leigh Abbott: Writing – review & editing, Visualization, Methodology, Conceptualization, Software, Data curation, Writing – original draft, Validation, Formal analysis. **Milan Marocchi:** Writing – original draft, Software, Writing – review & editing, Visualization, Validation, Conceptualization. **Matthew Fynn:** Writing – review & editing. **Yue Rong:** Writing – review & editing, Resources, Supervision, Project administration. **Sven Nordholm:** Writing – review & editing, Resources, Supervision, Project administration.

Ethics approval and consent

The study received approval from the ethics committee of Fortis Hospital, Kolkata, India, where the multichannel data collection occurred. Informed consent was obtained from all participating subjects. All other datasets are open-access, so no approval is required.

Declaration of competing interest

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

Acknowledgement

We thank *Ticking Heart Pty Ltd* for allowing the use of data collected from their vest design. We also thank Harry Walters for his valuable remarks and feedback on this work.

Appendix A. Conditional denoising diffusion probabilistic model equations

DDPMs considers the conditional distribution $p_\theta(\mathbf{y}_0|\mathbf{x})$, with $\mathbf{y}_0$ being the original waveform and $\mathbf{x}$ the conditioning features that correspond with $\mathbf{y}_0$,

$$p_\theta\left(\mathbf{y}_0|\mathbf{x}\right) = \int p_\theta\left(\mathbf{y}_{0:N}|\mathbf{x}\right) d\mathbf{y}_{1:T} \tag{A.1}$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_T$ is a series of latent variables. The posterior $q\left(\mathbf{y}_{1:T}|\mathbf{y}_0\right)$ is the forward diffusion process, which is defined through the Markov chain:

$$q\left(\mathbf{y}_{1:T}|\mathbf{y}_0\right) = \prod_{t=1}^{T} q\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right) \tag{A.2}$$

Gaussian noise being added in each iteration is defined as,

$$q\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right) = \mathcal{N}\left(\mathbf{y}_t; \sqrt{1-\beta_t}\mathbf{y}_{t-1}, \beta_t I\right) \tag{A.3}$$

with the noise being defined with a fixed noise schedule for $\beta_1, \ldots, \beta_T$. Hence, the diffusion process can be computed for any t as

$$\mathbf{y}_t = \sqrt{\overline{\alpha}_t}\mathbf{y}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_t \tag{A.4}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha_t} = \prod_{i=1}^{t} \alpha_i$. As the likelihood in Eq. (A.1) is intractable, training these models is done by maximising its variational lower bound (ELBO). Ho et al. [39] found that using a loss as defined in Eq. (A.5) leads to higher quality generation.

$$\mathbb{E}_{t,\epsilon}\left[\left\|\epsilon_\theta\left(\mathbf{y}_t, \mathbf{x}, t\right) - \epsilon_t\right\|_2^2\right] \tag{A.5}$$

The model estimates the noise added in the forward process, which is written as ϵ_θ and the actual noise added is written as ϵ_t, where $\epsilon_t \sim \mathcal{N}(0, I)$.

Table B.11
Training hyperparameters for the cβ-VAE model.

Hyperparameter	Value
Initial learning rate	2×10^{-4}
Batch size	8
Latent dimension size	32
Label embedding dimension size	32
ResBlock feature sizes (blocks 1–3)	32, 64, 128
ResBlock strides (blocks 1–3)	2, 2, 2
α (reconstruction loss weight)	1.0
θ (STFT loss weight)	0.5

Generation is then done by first sampling $\mathbf{y}_T \sim \mathcal{N}(0, I)$ and $\mathbf{z} \sim \mathcal{N}(0, I)$, before following the below equation until for $t = T, \ldots, 1, 0$,

$$\mathbf{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{y}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta\left(\mathbf{y}_t, \mathbf{x}, t\right)\right) + \sigma_t \mathbf{z} \tag{A.6}$$

where $\sigma_t = \tilde{\beta}_t$ and $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$ is the variance at step t for $t > 1$ and $\tilde{\beta}_1 = \beta_1$.

Appendix B. Cβ-VAE architecture and training

The cβ-VAE architecture is shown in Fig. B.13. The label and conditioner encoders are single 1D convolutional layers designed to align the label embedding and conditioning features with the feature dimensions of the layers to which they are added.

Each residual block (ResBlock) consists of two 1D convolutional layers, each followed by batch normalisation. In the encoder, the first convolution in a ResBlock downsamples the signal via striding, while the second maintains the same number of input and output channels. Both layers use a kernel size of 3; the first layer has a variable stride for downsampling, while the second has a fixed stride of 1.

The ResBlocks in the decoder do not perform any upsampling or downsampling. They use the same number of input and output channels for both layers, with all other parameters mirroring those in the encoder. The dimensions and sizes of layers are in Table B.11.

The model was trained for 24 h on an RTX 3090 GPU using the same preprocessing steps and collator as the diffusion models. Training used the AdamW optimiser with a learning rate scheduler that reduces the learning rate on loss plateau.

The total loss function, shown in Eq. (B.1), combines signal reconstruction loss, KL divergence, and multiscale STFT loss. Here, Z_μ and Z_σ are the encoder's outputs representing the latent mean and standard deviation, respectively:

$$\mathcal{L} = \beta D_{\mathrm{KL}}\left(\mathcal{N}(Z_\mu, Z_\sigma)|\mathcal{N}(0,1)\right) + \alpha\mathcal{L}_{\mathrm{recon}} + \theta\mathcal{L}_{\mathrm{mSTFT}} \tag{B.1}$$

The reconstruction loss, $\mathcal{L}_{\mathrm{recon}}$, is defined as the mean squared error (MSE) between the reference signal $\mathbf{s}_{\mathrm{ref}}$ and the generated signal $\mathbf{s}_{\mathrm{gen}}$:

$$\mathcal{L}_{\mathrm{recon}} = \left\|\mathbf{s}_{\mathrm{ref}} - \mathbf{s}_{\mathrm{gen}}\right\|_2^2 \tag{B.2}$$

The multiscale STFT loss, $\mathcal{L}_{\mathrm{mSTFT}}$, is computed by comparing the STFTs of the reference and generated signals using four different window sizes: $w \in \{256, 512, 1024, 2048\}$. Each comparison uses L1 loss, and the results are averaged:

$$\mathcal{L}_{\mathrm{mSTFT}} = \frac{1}{|w|}\sum_{w}\left\|\mathbf{S}_{\mathrm{ref},w} - \mathbf{S}_{\mathrm{gen},w}\right\|_1 \tag{B.3}$$

The KL divergence weight β was scheduled to increase during training, starting at 0.1 and linearly ramping up to 1.0 over 20,000 steps.

The training hyperparameters are summarised in Table B.11.

Appendix C. cDCGAN architecture and training

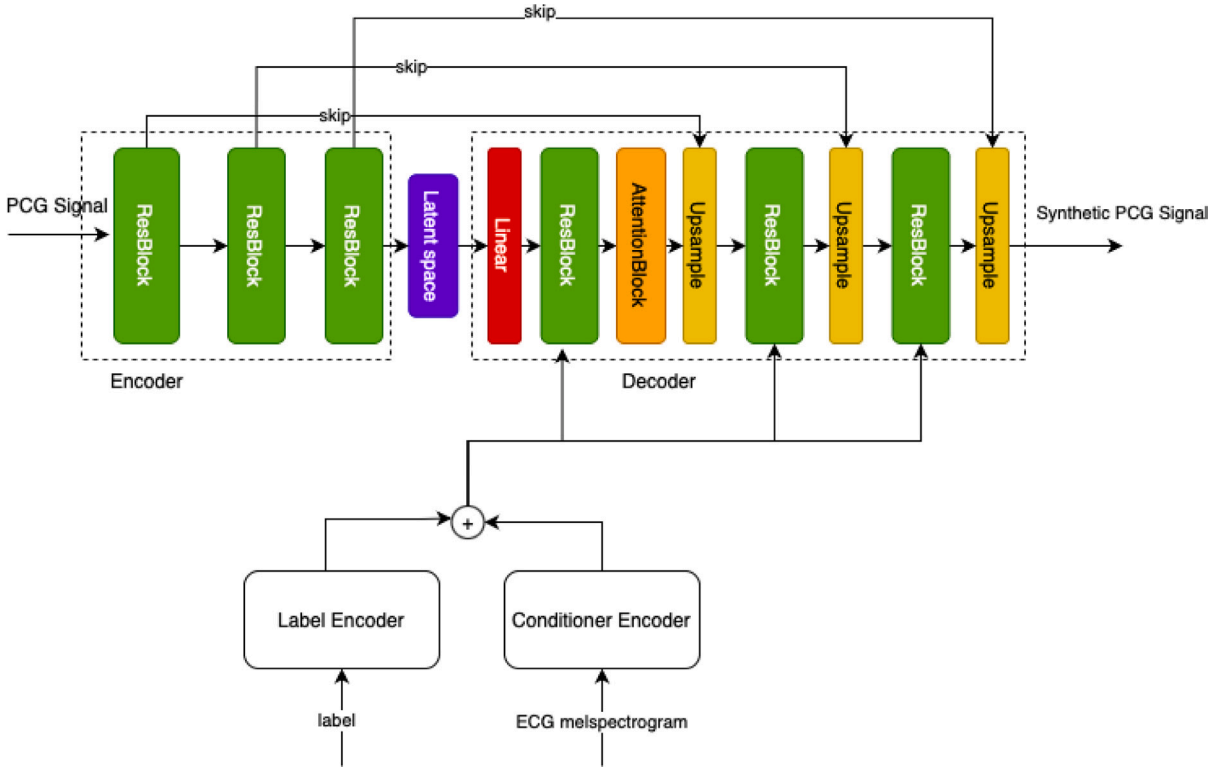


Fig. B.13. cβ-VAE architecture diagram.

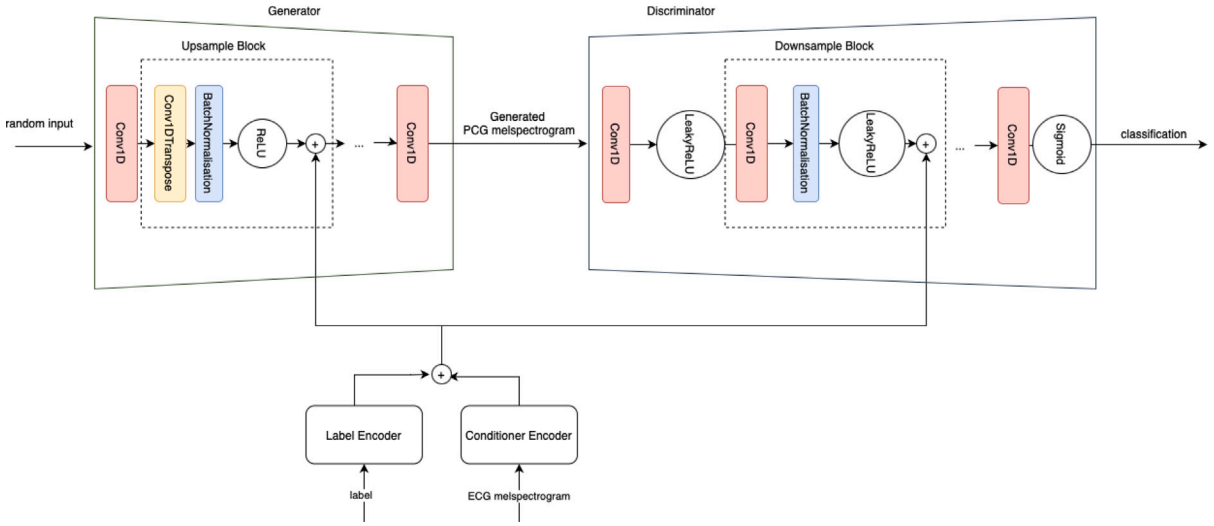
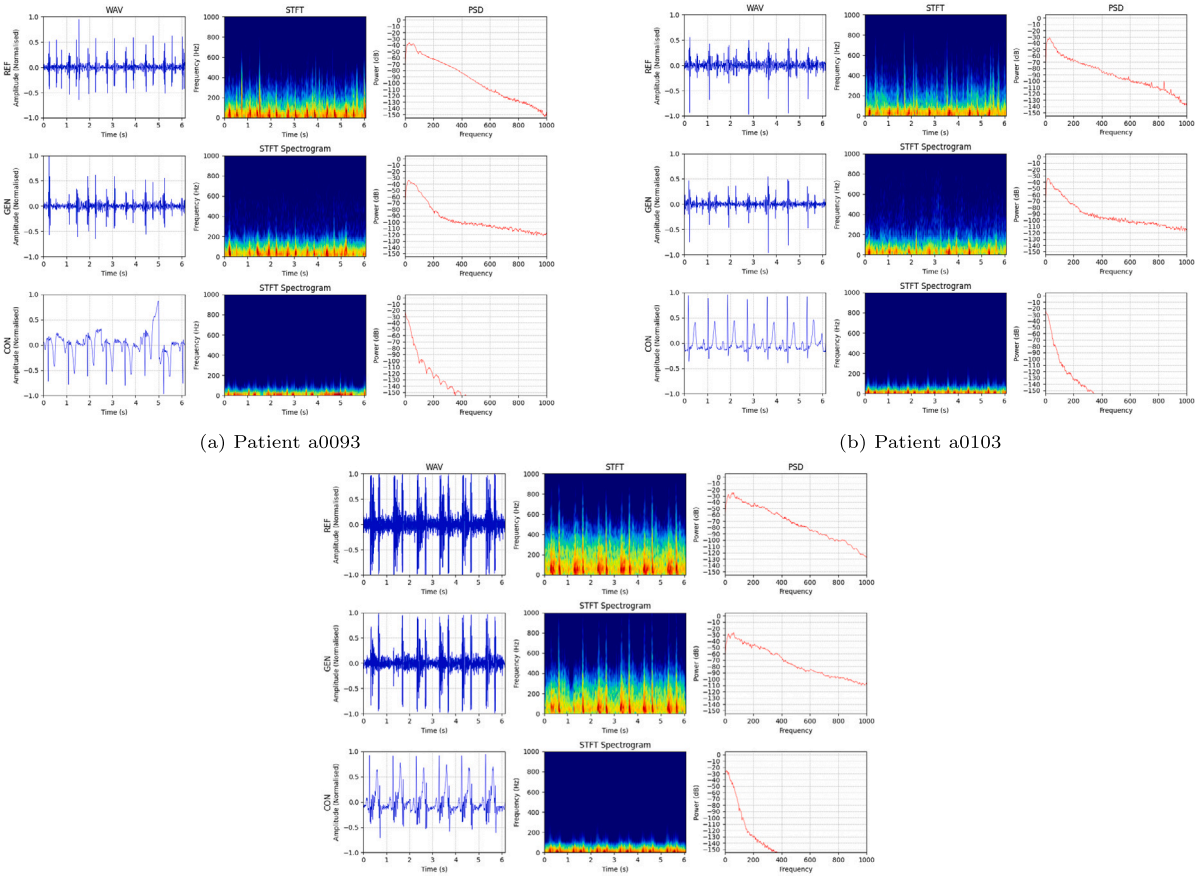


Fig. C.14. cDCGAN architecture diagram.

The cDCGAN architecture is shown in Fig. C.14. The label and conditioner encoders are single 1D convolutional layers designed to align the label embedding and conditioning features with the feature dimensions of the layers to which they are added. The size of the layers and number of layers are shown in Table C.12. The model was trained for 24 h on an RTX 3090 GPU using the same preprocessing steps and collator as the diffusion models. Training used the AdamW optimiser with a learning rate scheduler that reduces the learning rate on loss plateau.

The total loss function shown in Eq. (C.1), combines the adversarial loss with signal reconstruction loss, multiscale STFT loss and feature matching loss.

$$\mathcal{L} = \beta\mathcal{L}_{\text{Adv}} + \alpha\mathcal{L}_{\text{recon}} + \theta\mathcal{L}_{\text{mSTFT}} + \lambda\mathcal{L}_{FM} \tag{C.1}$$



(a) Patient a0093

(b) Patient a0103

(c) Patient a0400

Fig. D.15. Generated DiffWave signals for three different patients.

Table C.12
Training hyperparameters for the cDCGAN model.

Hyperparameter	Value
Initial learning rate	2×10^{-4}
Batch size	8
Latent dimension (random input) dimension	32
Generator/Discriminator feature dimension size	64
Upsample/Downsample scales	4, 2, 2, 6
α (reconstruction loss weight)	1.0
β (adversarial loss weight)	1.0
θ (STFT loss weight)	5.0
λ (feature matching loss weight)	5.0

The adversarial loss is defined as the sum of the discriminator and generator losses as in Eq. (C.2).

$$\mathcal{L}_{Adv} = \mathcal{L}_D + \mathcal{L}_G \tag{C.2}$$

The discriminator loss is described as follows,

$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x,y)\right] - \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z,y),y)\right)\right] \tag{C.3}$$

The generator loss is found in Eq. (C.4).

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z)}\left[\log D(G(z,y),y)\right] \tag{C.4}$$

where:

- $x \sim p_{\text{data}}(x)$: Real data sampled from the true data distribution.
- $z \sim p_z(z)$: Latent vector sampled from Gaussian noise (random input).
- y: Conditional input (ECG mel spectrogram).
- $G(z, y)$: Generator output conditioned on random input z and conditional input y.
- $D(x, y)$: Discriminator's estimate of the probability that x is real, given condition y.
- $\mathcal{L}_D$: Discriminator loss—maximised to distinguish real from fake samples.
- $\mathcal{L}_G$: Generator loss—minimised to fool the discriminator.

The reconstruction loss and multiscale STFT loss are the same that were utilised within the cβ-VAE model in Eq. (B.2) and Eq. (B.3).

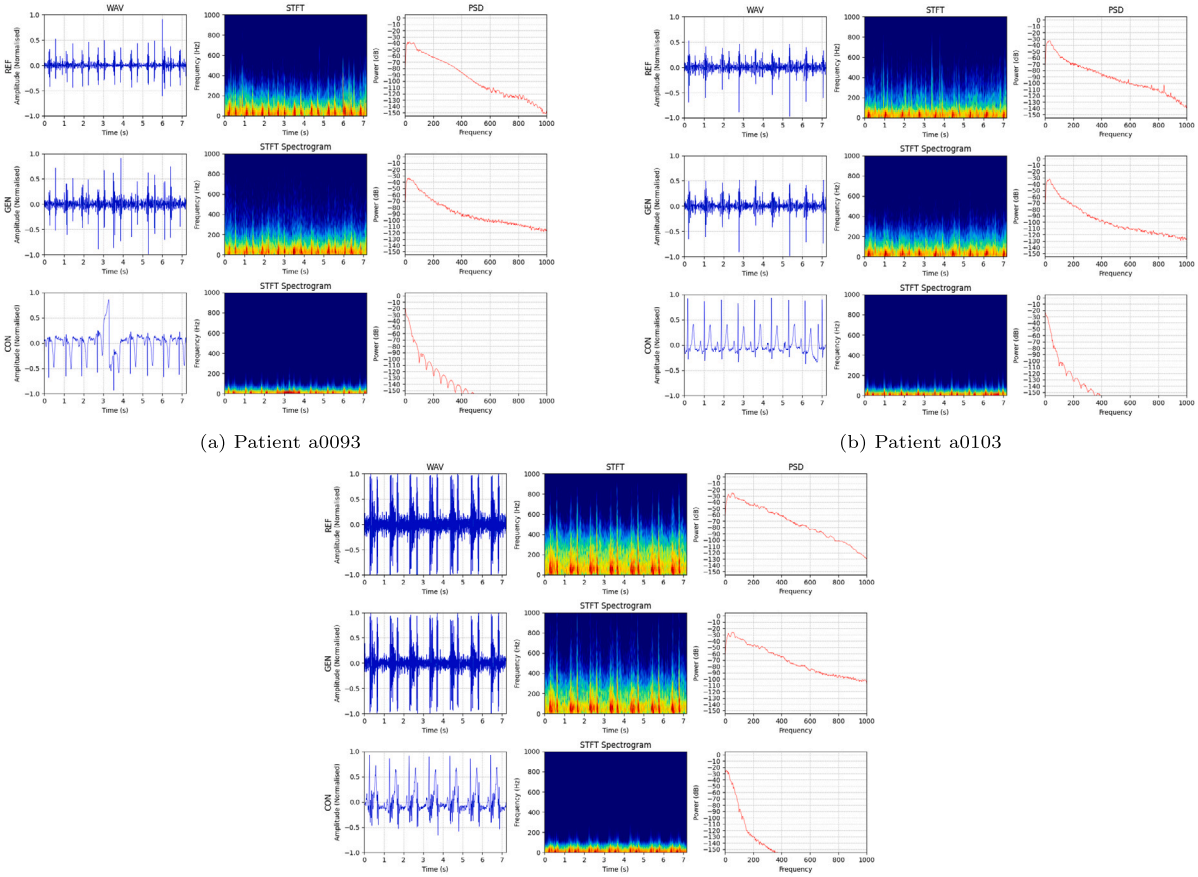
Lastly, the feature matching loss is defined as the L1 loss between the features, embedding taken from the last downsample block, of the discriminator for the reference signal and the generated signal, where F_{ref} are the features of the reference signal F_{gen} are the features of the generated signal.

$$\mathcal{L}_{FM} = \left\| F_{ref} - F_{gen} \right\|_1 \tag{C.5}$$

The training hyperparameters are summarised in Table C.12.

Appendix D. Diffusion model generation examples

See Figs. D.15 and D.16.



(a) Patient a0093

(b) Patient a0103

(c) Patient a0400

Fig. D.16. Generated WaveGrad signals for three different patients.

Data availability

Data will be made available on request.

References

[1] Cardiovascular diseases (CVDs), URL https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[2] T.R. Reed, N.E. Reed, P. Fritzson, Heart sound analysis for symptom detection and computer-aided diagnosis, Simul. Model. Pr. Theory 12 (2) (2004) 129–146, URL https://linkinghub.elsevier.com/retrieve/pii/S1569190X04000206.

[3] Y. Rong, M. Fynn, S. Nordholm, S. Siaw, G. Dwivedi, Wearable electro-phonocardiography device for cardiovascular disease monitoring, 2023, URL https://ddfe.curtin.edu.au/yurong/SSP23.pdf.

[4] C. Thomae, A. Dominik, Using deep gated RNN with a convolutional front end for end-to-end classification of heart sound, in: 2016 Computing in Cardiology Conference, CinC, 2016, pp. 625–628.

[5] Z. Kong, W. Ping, J. Huang, K. Zhao, B. Catanzaro, DiffWave: A versatile diffusion model for audio synthesis, 2021.

[6] N. Chen, Y. Zhang, H. Zen, R.J. Weiss, M. Norouzi, W. Chan, WaveGrad: Estimating gradients for waveform generation, 2020.

[7] M. Marocchi, L. Abbott, Y. Rong, S. Nordholm, G. Dwivedi, Abnormal heart sound classification and model interpretability: A transfer learning approach with deep learning, J. Vasc. Dis. 2 (4) (2023) 438–459, URL https://www.mdpi.com/2813-2475/2/4/34.

[8] C. Liu, D. Springer, Q. Li, B. Moody, R.A. Juan, F.J. Chorro, F. Castells, J.M. Roig, I. Silva, A.E.W. Johnson, Z. Syed, S.E. Schmidt, C.D. Papadaniil, L. Hadjileontiadis, H. Naseri, A. Moukadem, A. Dieterlen, C. Brandt, H. Tang, M. Samieinasab, M.R. Samieinasab, R. Sameni, R.G. Mark, G.D. Clifford, An open access database for the evaluation of heart sound algorithms, Physiol. Meas. 37 (12) (2016) 2181–2213, Cited by: 382; All Open Access, Green Open Access, URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85007492920&doi=10.1088%2f0967-3334%2f37%2f12%2f2181&partnerID=40&md5=bf943637cf0ed9217d6f4243debcde9c.

[9] A. Leatham, Auscultation of the Heart and Phonocardiography, 2nd, Churchill Livingstone, Edinburgh and New York, 1975.

[10] S.E. Schmidt, C. Holst-Hansen, J. Hansen, E. Toft, J.J. Struijk, Acoustic Features for the Identification of Coronary Artery Disease, IEEE Trans. Biomed. Eng. 62 (11) (2015) 2611–2619.

[11] D.B. Springer, L. Tarassenko, G.D. Clifford, Logistic regression-HSMM-based heart sound segmentation, IEEE Trans. Biomed. Eng. 63 (4) (2016) 822–832.

[12] R. Rajni, I. Kaur, Electrocardiogram signal analysis-an overview, Int. J. Comput. Appl. 84 (7) (2013) 22–25.

[13] G.D. Clifford, F. Azuaje, P. McSharry, Advanced Methods and Tools for ECG Data Analysis, Artech House, 2006.

[14] C. Xie, Biomedical signal processing: An ECG application, in: L.A. Celi, M.S. Majumder, P. Ordóñez, J.S. Osorio, K.E. Paik, M. Somai (Eds.), Leveraging Data Science for Global Health, Springer International Publishing, Cham, 2020, pp. 285–303, http://dx.doi.org/10.1007/978-3-030-47994-7_17.

[15] D. De Bacquer, G. De Backer, M. Kornitzer, K. Myny, Z. Doyen, H. Blackburn, Prognostic value of ischemic electrocardiographic findings for cardiovascular mortality in men and women, J. Am. Coll. Cardiol. 32 (3) (1998) 680–685.

[16] D. Tran, J. Liu, M.W. Dusenberry, D. Phan, M. Collier, J. Ren, K. Han, Z. Wang, Z. Mariet, H. Hu, N. Band, T.G.J. Rudner, K. Singhal, Z. Nado, J. van Amersfoort, A. Kirsch, R. Jenatton, N. Thain, H. Yuan, K. Buchanan, K. Murphy, D. Sculley, Y. Gal, Z. Ghahramani, J. Snoek, B. Lakshminarayanan, Plex: Towards reliability using pretrained large model extensions, 2022, URL https://storage.googleapis.com/plex-paper/plex.pdf.

[17] E. Briscoe, J. Feldman, Conceptual complexity and the bias/variance tradeoff, Cognition 118 (1) (2011) 2–16, http://dx.doi.org/10.1016/j.cognition.2010.10.004.

[18] H. Yao, Y. Wang, S. Li, L. Zhang, W. Liang, J. Zou, C. Finn, Improving out-of-distribution robustness via selective augmentation, 2022, URL https://arxiv.org/abs/2201.00299.

[19] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437, URL https://www.sciencedirect.com/science/article/pii/S0306457309000259.

[20] I.M. De Diego, A.R. Redondo, R.R. Fernández, J. Navarro, J.M. Moguerza, General performance score for classification problems, Appl. Intell. 52 (10) (2022) 12049–12063, http://dx.doi.org/10.1007/s10489-021-03041-7.

[21] Z. Ren, Y. Chang, T.T. Nguyen, Y. Tan, K. Qian, B.W. Schuller, A comprehensive survey on heart sound analysis in the deep learning era, 2023.
[22] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC Genomics 21 (6) (2020) http://dx.doi.org/10.1186/s12864-019-6413-7.
[23] M. Ding, K. Kong, J. Chen, J. Kirchenbauer, M. Goldblum, D. Wipf, F. Huang, T. Goldstein, A closer look at distribution shifts and out-of-distribution generalization on graphs, in: NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications, 2021, URL https://openreview.net/forum?id=XvgPGWazqRH.
[24] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, 2021, URL http://arxiv.org/abs/2003.05991.
[25] D.P. Kingma, M. Welling, An introduction to variational autoencoders, Found. Trends® Mach. Learn. 12 (4) (2019) 307–392, http://dx.doi.org/10.1561/2200000056.
[26] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, URL http://arxiv.org/abs/1406.2661.
[27] J. Sohl-Dickstein, E.A. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, 2015, URL http://arxiv.org/abs/1503.03585.
[28] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, M.-H. Yang, Diffusion models: A comprehensive survey of methods and applications, 2023, URL https://arxiv.org/abs/2209.00796.
[29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, 2022, URL http://arxiv.org/abs/2112.10752.
[30] Z. Xiao, K. Kreis, A. Vahdat, Tackling the generative learning trilemma with denoising diffusion GANs, 2022, URL http://arxiv.org/abs/2112.07804.
[31] W.H.L. Pinaya, P.-D. Tudosiu, J. Dafflon, P.F. da Costa, V. Fernandez, P. Nachev, S. Ourselin, M.J. Cardoso, Brain imaging generation with latent diffusion models, 2022, URL http://arxiv.org/abs/2209.07162.
[32] G. Zhou, Y. Chen, C. Chien, On the analysis of data augmentation methods for spectral imaged based heart sound classification using convolutional neural networks, BMC Med. Inform. Decis. Mak. (2022).
[33] J. Saldanha, S. Chakraborty, S. Patil, K. Kotecha, S. Kumar, A. Nayyar, Data augmentation using variational autoencoders for improvement of respiratory disease classification, PLoS One 17 (8) (2022) 1–41, http://dx.doi.org/10.1371/journal.pone.0266467.
[34] K. Kochetov, A. Filchenkov, Generative adversarial networks for respiratory sound augmentation, in: Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System, CCRIS '20, Association for Computing Machinery, New York, NY, USA, 2021, pp. 106–111, http://dx.doi.org/10.1145/3437802.3437821.
[35] Z. Zhang, J. Han, K. Qian, C. Janott, Y. Guo, B. Schuller, Snore-GANs: Improving automatic snore sound classification with synthesized data, IEEE J. Biomed. Heal. Inform. 24 (1) (2020) 300–310.
[36] P. Narváez, W.S. Percybrooks, Synthesis of normal heart sounds using generative adversarial networks and empirical wavelet transform, Appl. Sci. 10 (19) (2020) URL https://www.mdpi.com/2076-3417/10/19/7003.
[37] T. Dissanayake, T. Fernando, S. Denman, S. Sridharan, C. Fookes, Generalized generative deep learning models for biosignal synthesis and modality transfer, IEEE J. Biomed. Heal. Inform. 27 (2) (2023) 968–979, http://dx.doi.org/10.1109/JBHI.2022.3223777.
[38] A. Kebaili, J. Lapuyade-Lahorgue, S. Ruan, Deep learning approaches for data augmentation in medical imaging: A review, J. Imaging 9 (4) (2023) URL https://www.mdpi.com/2313-433X/9/4/81.
[39] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, 2020.
[40] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L.C. Cobo, K. Simonyan, High fidelity speech synthesis with adversarial networks, 2019.
[41] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A generative model for raw audio, 2016, URL https://arxiv.org/abs/1609.03499.
[42] G.D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, R.G. Mark, Classification of normal/abnormal heart sound recordings: The PhysioNet/Computing in cardiology challenge 2016, Comput. Cardiol. 43 (2016) 609–612, Cited by: 128, URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85016097943&partnerID=40&md5=7c41089fc1564d3091a220eeaf126379.
[43] Y. Rong, M. Fynn, S. Nordholm, A pre-screening technique for coronary artery disease with multi-channel phonocardiography and electrocardiography, in: A. Al-Jumaily, P. Crippa, A. Mansour, C. Turchetti (Eds.), Non-Invasive Health Systems Based on Advanced Biomedical Signal and Image Processing, Taylor & Francis, 2023, URL https://www.taylorfrancis.com/books/edit/10.1201/9781003346678/non-invasive-health-systems-based-advanced-biomedical-signal-image-processing-adel-al-jumaily-paolo-crippa-ali-mansour-claudio-turchetti.
[44] S. Tan, G. Androz, A. Chamseddine, P. Fecteau, A. Courville, Y. Bengio, J.P. Cohen, Icentia11K: An unsupervised representation learning dataset for arrhythmia subtype discovery, 2019.
[45] A. Kazemnejad, P. Gordany, R. Sameni, EPHNOGRAM: A simultaneous electrocardiogram and phonocardiogram database, 2021.
[46] G. Moody, R. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50.
[47] B. McFee, librosa/librosa: 0.10.1, 2023, http://dx.doi.org/10.5281/zenodo.8252662, (Accessed 24 March 2024).
[48] J. Driedger, M. Muller, S. Disch, Extending harmonic-percussive separation of audio signals, in: Proceedings of the International Society for Music Information Retrieval Conference, vol. 15, 2014.
[49] R. Bristow-Johnson, A theory of optimal splicing of audio in the time domain, 2011, Music-DSP Mailing List, URL https://music.columbia.edu/pipermail/music-dsp/2011-July/069971.html. (Accessed 24 March 2024).
[50] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272.