# Pulmonary Hypertension Detection From Heart Sound Analysis

Alex Gaudio, Noemi Giordano, *Member, IEEE*, Mounya Elhilali, *Senior Member, IEEE*, Samuel Schmidt, and Francesco Renna

***Abstract*—The detection of Pulmonary Hypertension (PH) from the computer analysis of digitized heart sounds is a low-cost and non-invasive solution for early PH detection and screening. We present an extensive cross-domain evaluation methodology with varying animals (humans and porcine animals) and varying auscultation technologies (phonocardiography and seisomocardiography) evaluated across four methods. We introduce PH-ELM, a resource-efficient PH detection model based on the extreme learning machine that is smaller ($300\times$ fewer parameters), energy efficient ($532\times$ fewer watts of power), faster ($36\times$ faster to train, $44\times$ faster at inference), and more accurate on out-of-distribution testing (improves median accuracy by 0.09 area under the ROC curve (auROC)) in comparison to a previously best performing deep network. We make four observations from our analysis: (a) digital auscultation is a promising technology for the detection of pulmonary hypertension; (b) seismocardiography (SCG) signals and phonocardiography (PCG) signals are interchangeable to train PH detectors; (c) porcine heart sounds in the training data can be used to evaluate PH from human heart sounds (the PH-ELM model preserves 88 to 95% of the best in-distribution baseline performance); (d) predictive performance of PH detection can be mostly preserved with as few as 10 heartbeats and capturing up to approximately 200 heartbeats per subject can improve performance.**

***Index Terms*—Auscultation, biomedical signal analysis, pulmonary hypertension, machine learning.**

## I. Introduction

PULMONARY Hypertension (PH) is a hemodynamic condition characterized by an increased pulmonary artery pressure (PAP) and an increased afterload in the right ventricle [1]. PH affects both the cardiac and the pulmonary functionality, it can co-occur with several cardiovascular and respiratory diseases [2], including heart failure, and it is associated with increased mortality [3]. Its prevalence was estimated to affect 1% of the global population [2], [4].

Two main challenges in PH screening include: (a) the need for reliable, low-cost, and non-invasive technology; and (b) the non-specific symptomatic presentation of the condition [3], [5]. While delayed PH diagnosis has been linked with decreased survival rate [6], [7], guidelines for PH diagnosis are designed to balance the benefits of early detection with the economic healthcare burden that early screening places on PH referral centers [5]. The diagnosis of PH, according to recent guidelines from the European Society of Cardiology (ESC) and European Respiratory Society (ERS) [2], is obtained via the right heart catheterization (RHC). The RHC is an invasive and expensive procedure to measure the systolic, diastolic, and mean pulmonary artery pressure (PAP), and it is therefore not suitable for screening.

The clinical need for reliable and non-invasive technologies to raise early suspicion of PH [2] is not well met by current technological capabilities. A reliable, low-cost, and non-invasive approach for raising suspicion of PH can reduce risks and costs, and improve awareness of PH in telemedicine, screening settings, developing countries, rural areas, and underprivileged settings. Existing technologies are used together to raise suspicion of PH and justify performing a final diagnosis by RHC [8]. These technologies include electrocardiography (EKG), blood tests, echocardiography (ECHO), chest radiology, magnetic resonance imaging (MRI), and pulmonary function tests such as spirometry and arterial blood gas [2], [5]. The 2022 ESC/ERS guidelines recommend ECHO as a main tool for the detection of PH, to be performed after a thorough physical exam including blood tests and resting EKG, and before final confirmation with RHC. The ECHO technology, however, has important drawbacks: (a) no single ECHO biomarker reliably informs about PH status [2] and it cannot estimate pulmonary pressure in 10-50% of patients [9]; (b) a normal ECHO does not exclude PH [5]; and (c) the ECHO procedure is not low resource cost because it requires a trained sonographer, an analysis by a cardiologist and typically also a referral to an imaging center or hospital.

Received 8 September 2024; revised 7 February 2025; accepted 25 March 2025. Date of publication 28 March 2025; date of current version 15 September 2025. This work was supported by the National Institutes of Health under Grant 1R01HL163439. The work of Francesco Renna was supported by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within Project HfPT, with reference 41. *(Corresponding author: Alex Gaudio.)*

Alex Gaudio is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: agaudio2@jh.edu).

Mounya Elhilali is with the Department of Electrical and Computer Engineering, Johns Hopkins University USA.

Noemi Giordano is with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy.

Samuel Schmidt is with the Faculty of Medicine, University of Aalborg, Denmark.

Francesco Renna is with the INESC TEC, Faculdade de Ciências da Universidade do Porto, Portugal.

Digital Object Identifier 10.1109/TBME.2025.3555549

We propose an under-recognized but promising technology for raising suspicion of PH: digital auscultation and the computer analysis of heart sounds.

*Contributions:* This work contributes a PH-ELM predictive model for PH detection from heart sounds. The model has a very low resource footprint and higher prediction performance in out-of-distribution (OOD) tests when compared to competing heart sound analysis approaches. We also contribute an extensive methodology and analysis for the in-distribution and out-of-distribution testing of three heart sound datasets and four predictive models. We study PH detection in different and changing data modalities, including heart sounds recorded with seismocardiography (SCG) and phonocardiography (PCG) technologies, and heart sounds from humans and pigs. For the science of heart sound auscultation in pulmonary hypertension, this work demonstrates that heart sounds auscultated via digital stethoscopes or via accelerometers can contain sufficient information for accurate PH detection. Moreover, we provide evidence that training datasets for PH detection can mix PCG and SCG signals or use either interchangeably, and that PH detectors can be trained on porcine heart sounds and evaluated on human heart sounds.

Section II reviews background material and surveys related works to contextualize and justify the proposed methodology. Section III describes the methods and datasets. Section IV shows results supporting our methodology and scientific claims. Section V offers insight into the scope of this work and the merits of the proposed modeling and evaluation methodologies.

## II. Background and Related Works

*Background:* Within each heartbeat, the closing of the heart's four valves creates two main sounds colloquially referred to as the "lub" and the "dub" sounds. The "lub" sound, known as the first heart sound, or S1, arises from the closures of the two atrioventricular valves [17]. The second heart sound (S2) [18], or the "dub" sound, occurs as a result of the closure of the aortic valve (A2) and pulmonic valve (P2). The time delay and relative intensity of the P2 are known to be relevant for pulmonary hypertension detection [19], while the third and fourth heart sounds, known as S3 [20] and S4, are often not audible by a human ear but they can be useful for pulmonary hypertension detection [10], [12].

*Related Works:* Some of the existing PH detection methods create hand-crafted features that extract information from the S1, S2, S3, and S4 components of a heart sound recording, and then evaluate the features using correlation, significance testing, or linear regression. Yamakawa et al. [10] studied intensity, complexity, and strength features extracted from each of the four fundamental heart sounds to show that S2 complexity and S3 intensity are the most statistically significant features for PH classification. They also obtained auROC values by training and evaluating the predictive performance of single features for different kinds of PH. Huang et al. [11] used reference PAP estimates obtained from ECHO to partition human subjects into three levels of PH (mild, medium, severe) and a control group. They found that: S2 amplitude and frequency alone were predictive of PH; ratios of the S2/S1 energy, amplitude, and frequency were all higher in subjects with PH; and the amplitude, frequency, and energy of S1 sounds are not correlated with ECHO readings. In 2010, Dennis et al. [12] utilized a variety of features on all four fundamental heart sounds and the authors report good results while investing minimal effort on manual pre-processing. Kaddoura et al. [13] applied a Gaussian Mixture Model to features extracted from the Mel-frequency Cepstral Coefficients (MFCC) of S2 segments. While their experiment was shown to outperform physician auscultation by a large margin, their relatively low overall performance (0.74 auROC) compared to the other works in Table I is possibly due to the choice of features. MFCC features are designed to approximate the human auditory system's response, but their result with human auscultators also suggests the human ear may not provide a good reference model for PH detection. While hand-crafted features can yield models with low computational resource requirements and useful interpretations, the predictive performance of models relying on these methods is often fundamentally limited by the availability of and ability to encode domain knowledge into the generated features. Our proposed fixed-weight approach bypasses limitations by randomly generating features and benefits from the associated computational improvements.

Data-driven machine learning techniques use a dataset to optimize model parameters, and they create or utilize features that are learned or automatically optimized to the data. Wang et al. [14] tested 10 different Deep Learning models on a set of heart sound features obtained from a magnitude spectrogram of the continuous wavelet transform and augmented with random noise. While the authors report 0.98 accuracy (averaged across multiple classifications including PH) and a similarly high F1 score for PH over a 10-fold cross-validation, they state that the experiment's dataset mixed the same subjects in the training and validation sets. When training data appears in the validation set, the empirical prediction performance is arguably over-optimistic due to overfitting. Ge et al. [16] extracted features from segmented S1 and S2 sounds by utilizing fixed-weight and data-driven methods, including hand-crafted features, time-frequency analysis with wavelets, and a convolutional neural network feature extractor. The study partitions 2415 recordings of 438 subjects into train, validation, and test sets. It was not specified if the subsets were stratified by subject. Our previous work [15], [19], [21] proposed a CNN-based PH detection method to evaluate an image representation of S2 segments. It incorporated a pre-processing method based on blind source separation to extract A2 and P2 waveforms from each S2 segment. The empirical analysis claimed that the availability of separated A2 and P2 components contributes significantly to prediction, but it was only cross-validated with one dataset, and 10-fold cross-validation performed only once.

## III. Materials and Methods

We propose a methodology for PH detection based on the extreme learning machine architecture that has the high prediction performance of a deep network with a small computational footprint. We present an evaluation approach that considers both in-distribution testing and out-of-distribution testing with two ranking metrics. Section III-A describes the

TABLE I
PH DETECTION METHODS

| Reference | Year | Data Analyzed | Ground Truth | Approach | Evaluation | Performance | Num Subjects |
|---|---|---|---|---|---|---|---|
| Yamakawa et al. [10] | 2022 | S1, S2, S3, S4 | RHC | ANOVA | Not Specified | 0.67 - 0.81 auROC | 40 |
| Huang et al. [11] | 2023 | S1, S2, S3, S4 | ECHO | LR | Not Specified | 0.78/ 0.83/ 0.88 auROC | 209 |
| Dennis et al. [12] | 2010 | S1, S2, S3, S4 | ECHO | NB | Train:Test | 0.78 auROC | 20:31 |
| Kaddoura et al. [13] | 2016 | S2 | RHC | MM | 5-fold CV | 0.74 auROC | 164 |
| Wang et al. [14] | 2022 | S1, S2, S3, S4 | Unspecified | DNN | 10-fold CV | 0.99 auROC | 74 |
| Gaudio et al. [15] | 2022 | S2 | RHC | DNN | 10-fold CV | 0.95 auROC | 42 |
| Ge et al. [16] | 2023 | S1, S2 | ECHO | XGBoost | Train:Val:Test | 0.93 auROC | 483 |

Approach: LR means Logistic Regression; NB is Naive Bayes; DNN is Deep Neural Network; MM is Mixture Model.
Evaluation: CV means cross-validation; Train:Test is a 1-fold CV with the validation set defined as the test set. No results in the table validated results with an out-of-distribution dataset for the detection of PH.
Performance: auROC means area under the receiver operating characteristic (ROC) curve.

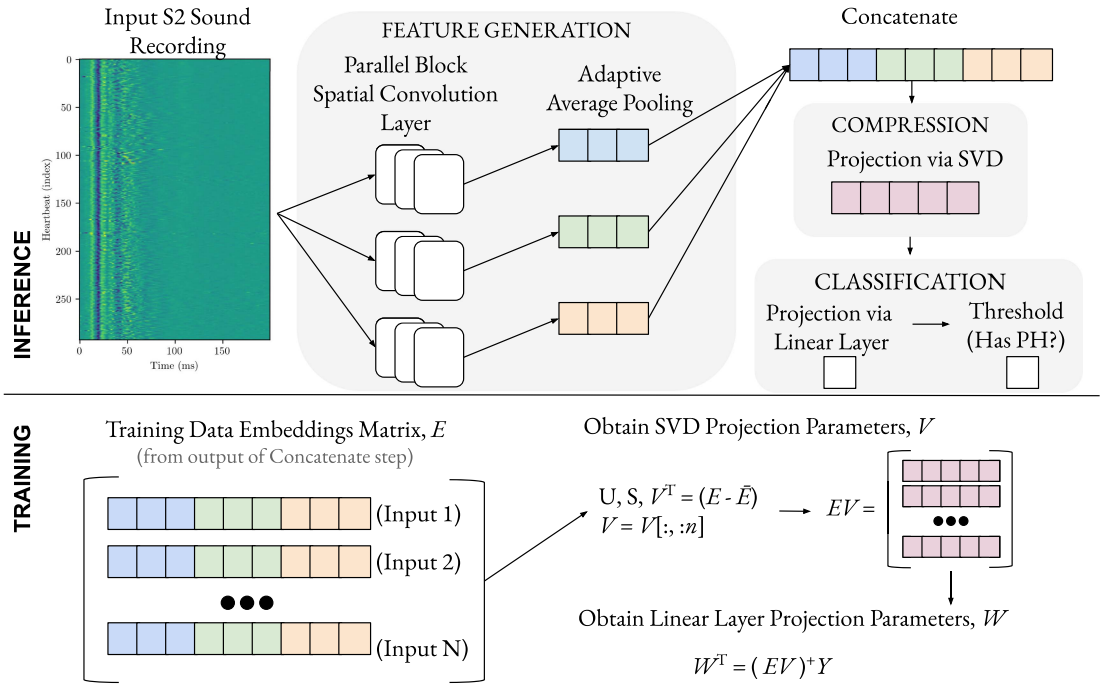


Fig. 1. **PH-ELM, an extreme learning machine model architecture with small computational footprint** designed with three parts: a fixed-weight, random and non-adaptive feature generation step, a compression step, and a classification utilizing ridge regression. Parallelized block convolutions, adaptive pooling, and singular value decomposition (SVD) convert each input heart sound recording into a compressed row vector. Training optimizes a SVD projection matrix and the weights for a final linear layer by direct analysis of a partially compressed representation of the training dataset.

proposed PH detector model. Section III-B describes the evaluation methodology, including an in-distribution analysis based on bootstrapped cross-validation and an out-of-distribution analysis across datasets. Section III-C describes the datasets used. Section III-D introduces the competing methods and hyperparameters.

### A. Proposed Model Based on Extreme Learning Machine

We restrict our focus to machine learning methods based on deep learning and the extreme learning machine. The model design, shown in Fig. 1, can be described as a resource-efficient extreme learning machine composed of three stages: feature generation, compression, and classification.

*Input:* The input is a pre-processed image matrix containing segmented S2 sounds, and it is visualized in Fig. 1. The image is obtained via the following analysis of a heart sound audio recording: a 200 ms time window vector of the recorded S2 heart sound is extracted by a dataset-specific segmentation, and then the ordered set of time window vectors are stacked as rows of a matrix. The output of the pre-processing step for each sample is a matrix with one row for each heartbeat and 200 columns that represent an aligned 200 ms window of the S2 sound. We assume each image is a single-channel grayscale image matrix, defined as $C = 1$ channel. Note that our prior works [15], [19] utilized this image structure and also introduced a $C = 3$ input image via a pre-processing step. The shape of the input is $(B, C, H, W)$ where $B = 1$ is the batch size,[1] $C \geq 1$ channels, $W = 200$ audio samples, and $H$ heartbeats is variable.

[1]Note that $B > 1$ implies the number of heartbeats is constant across images, or that all images are zero-padded or cropped to the same number of heartbeats. Zero-padding may introduce artifacts during feature generation due to the

*Feature Generation:* The feature generation step can be interpreted as a kind of time-frequency transform with a fixed set of random kernels and random fixed bias. A parallel block convolution layer converts the shape $(B, C, H, W)$ into a shape $(B, O, 1, 1)$ using a series of convolution layers, the ReLU nonlinear operator, and adaptive average pooling. The convolution layer evaluates $CO$ different $5 \times 5$ kernel matrices with a stride of 1 and dilation of 2 to attain a shape $(B, O, H', W')$, and it also uses a randomly initialized bias. The bias and kernel parameters are randomly initialized from uniform distributions with zero mean, and they are never modified after initialization. ReLU converts any negative values to zero. Adaptive Average Pooling converts $(B, O, H', W')$ to shape $(B, O, 1, 1)$. Because adaptive pooling converts the convolution feature matrix to a scalar number, it creates a feature generator with fixed output that accepts variable size input.

In the convolution layer, the kernel size and dilation can be matched to the sample rate of the represented signal of length $W$ . At a sample rate of 1000 Hz, a kernel size of $k = (5, 5)$ and dilation $d = (2, 2)$ creates a filter with an effective receptive field of $k + (k - 1)d = (13, 13)$ pixels, which defines a 13ms window over 13 consecutive heartbeats.

*Adjustable RAM utilization:* The parallel group convolution also introduces a set of $G$ groups, each of which computes $\frac{O}{G}$ outputs. Fig. 1 shows a set of $O$ white squares that are divided into $G$ groups. Each group of white squares is a single convolution layer with $\frac{O}{G}$ outputs. In this work, we set $O = 1000$ and $G = 10$, therefore each group (of white squares) is responsible for converting the input $(B, C, H, W)$ into a subset of the outputs $(B, O/G, H', W')$. Each group is computed in series, and by tuning $G$, the total amount of utilized memory is restricted; increasing $G$ decreases memory utilization and increases compute time.

*Compression and Classification:* The generated features are compressed by a principal components analysis and classified via ridge regression. The training phase and inference phase are subsequently described.

*Training:* The objective of the training phase is to solve a ridge regression on a compressed feature representation. In (1), the feature representation of each input training dataset image becomes a row of the matrix $E_{\text{CNN}}$, where $f(I_i)$ is the feature generation of the given input image. The columns of $E_{\text{CNN}}$ are reduced via SVD. We assume that zero centering by ensuring each column has zero mean does not affect downstream prediction performance because the bias and kernel parameters have zero mean in expectation.[2] The first $o$ columns of the projection matrix $V$ are used for compression and stored for inference on the validation set. The value $o$ is a hyperparameter that determines how many bases to use for compression, and we use $o = 20$.

$$E_{\text{CNN}} = \begin{bmatrix} \mathbf{f}(I_1) \\ \cdots \\ \mathbf{f}(I_N) \end{bmatrix} \in \mathbb{R}^{N \times O} \Rightarrow \begin{array}{c} U, S, V^\top = E_{\text{CNN}} - \bar{E}_{\text{CNN}} \\ V = V[:, : o] \\ E = E_{\text{CNN}} V \in \mathbb{R}^{N \times o} \end{array} \tag{1}$$

A ridge regression model generates a prediction of pulmonary hypertension with an objective defined in (2). The objective, defined in (2), does not include a bias term as a column of $E$ because it is assumed that the bias is zero in expectation, as a result of the feature generation and SVD. The Lagrange multiplier $\lambda = 0.1$ is chosen to avoid degenerate cases that could occur if a value on the diagonal of the square matrix $E^\top E$ equals zero, and $\mathbf{1}$ is an identity matrix.

$$\arg\min_{\mathbf{w}} ||\mathbf{y} - E\mathbf{w}||_2^2 + \lambda ||\mathbf{w}||_2^2 \tag{2}$$

$$\Rightarrow \mathbf{w} = (E^\top E + \lambda \mathbf{1})^{-1} E^\top y \tag{3}$$

*Binarization:* In the ground truth, if $y \in [0, 1]$, then a threshold on the predicted value $\hat{y} > 0.5$ is a reasonable classification boundary. If $y \in [-1, 1]$, then $\hat{y} > 0.0$ is a reasonable classification boundary. The threshold could also be found by standard analysis in ROC or PR space of a validation set (not a test set).

Training therefore introduces hyperparameter $o$, the number of columns of the compressed output, and stores learned parameters, $\mathbf{w} \in \mathbb{R}^{(o,1)}$, $\bar{\mathbf{E}} \in \mathbb{R}^{(1,O)}$ and $V \in \mathbb{R}^{(W,o)}$.

*Output Classification:* The scalar value $\hat{y} = (f(I) - \bar{\mathbf{E}})Vw$ determines an unnormalized score for PH detection that can be thresholded.

## B. Evaluation

Fig. 2 visually describes the evaluation methodology employed. This section first introduces the two rank metrics utilized for evaluation, then presents the in-distribution analysis methodology, and last presents the out-of-distribution analysis methodology and shows how we combine in-distribution with out-of-distribution analysis.

*Metrics for Empirical Evaluation:* We adopt the area under the ROC curve (auROC) and a variation of the area under the PR curve that we call AP*, described below and formally introduced in our concurrent work [23].

The auROC shows a change in score when the classifier is evaluated on datasets of varying imbalance ratios [24]. Moreover, the auROC is insensitive to model prediction balance, which means it penalizes all errors equally, regardless of whether the model outputs large prediction probabilities or small ones [25]. We adopt the standard implementation from the SciKit Learn library [26].

The AP* is designed to be sensitive to both class imbalance and model prediction imbalance because it penalizes prediction errors to the minority class more than errors to the majority. Given two models with identical auROC, the AP* gives a higher number to the model that is better at classifying the majority class, and it can therefore be useful in model selection. AP* is a variation of, and improvement upon, the average precision (AP). While the AP is non-symmetric in the presence of the

presence of a bias term in the feature generation convolution layers, and it is unknown how or if these artifacts impact performance.

[2]It as also found in [22] that zero centering paired with random CNN initialization did not meaningfully affect performance. However, our final implementation used zero centering.

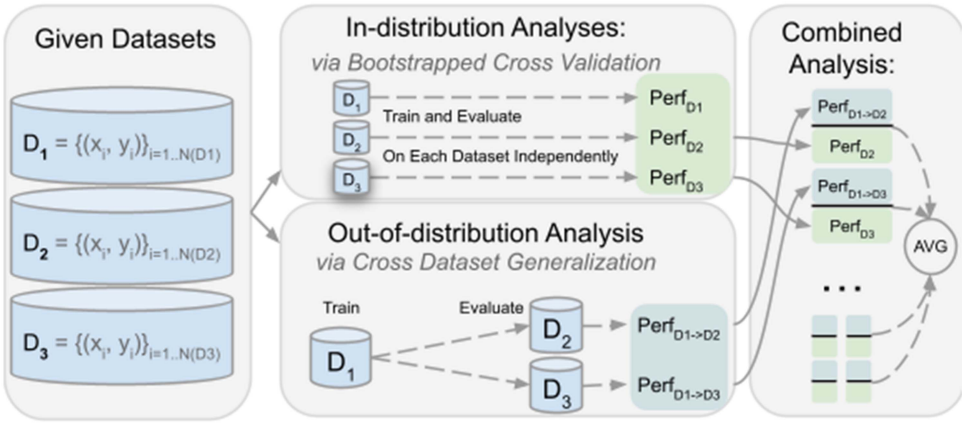


Fig. 2. Evaluation Methodology with in-distribution and out-of-distribution testing.

class imbalance, the AP* and the auROC are symmetric. The AP* ranges from [0,1], where the random classifier, all ones classifier, and all zeros classifier all have the minimum value of AP* $= \frac{\text{num samples in minority class}}{\text{dataset size}} \in [0, 0.5]$. It is defined as:

$$\mathbf{p}^+, \mathbf{r}^+, \mathbf{t} = \texttt{PR\_curve}(\mathbf{y}, \hat{\mathbf{y}}) \tag{4}$$

$$\mathbf{p}^-, \mathbf{r}^-, \mathbf{t} = \texttt{PR\_curve}(1 - \mathbf{y}, 1 - \hat{\mathbf{y}}) \tag{5}$$

$$\mathbf{p}^*, \mathbf{r}^* = \texttt{emin}(\mathbf{p}^+, \mathbf{p}^-), \texttt{emin}(\mathbf{r}^+, \mathbf{r}^-) \tag{6}$$

$$\texttt{AP}^*(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i \in \{1,\ldots,|T|\}} \left(r_i^* - r_{i-1}^*\right) p_i^* \tag{7}$$

where $\mathbf{y} \in \{0,1\}^n$ is a vector of the $n$ ground truth PH annotation labels and $\hat{\mathbf{y}} \in \mathbb{R}^n$ is a corresponding vector of the model's predictions. The function $\texttt{emin}(\cdot,\cdot)$ computes the element-wise minimum between two vectors. The vectors $\mathbf{p}^+, \mathbf{r}^+, \mathbf{p}^-, \mathbf{r}^-$ contain the precisions and recalls computed at each of the thresholds (in vector $\mathbf{t}$) emitted by a precision-recall curve function on the defined input. If (7) used $\mathbf{p}^+$ and $\mathbf{r}^+$ instead of $\mathbf{p}^*$ and $\mathbf{r}^*$, it would be identical to the average precision score presented in the SciKit Learn library [26] and [27]. Whereas the average precision score assigns a larger penalty when the prediction probability is large, the AP* assigns a larger penalty to the minority class. A detailed introduction to the AP* is described in concurrent work [23].

*In-distribution Analysis Methodology: Bootstrapped Cross-Validation* describes model performance using one dataset. On small datasets, such as all datasets used in this paper, cross-validation analysis (CV) alone may be insufficient because the way the folds are generated may have a significant impact on model performance. Bootstrapped cross-validation improves the reliability of the reported performance. By bootstrapping, we mean to compute $K$-fold CV multiple times independently, where each time, the folds are randomly generated using a different random seed. In the special cases of cross-validation with no variation in the folds, such as group cross-validation where each group is fixed, bootstrapping can capture possible random variations in model initialization or training. The following notation is used: in a given dataset $D$, we define each $k^{\text{th}}$ fold, $k \in \{1, \ldots, K\}$ and each $b^{\text{th}}$ bootstrap iteration, $b \in \{1, \ldots, B\}$. For each fold and bootstrap iteration, we have two disjoint sets: the training set $D_{b,k,\text{train}} \subset D$ and validation set $D_{b,k,\text{val}} \subset D$. After training the model on $D_{b,k,\text{train}}$, the evaluation on $D_{b,k,\text{val}}$ yields a ground truth vector $\mathbf{y}_{k,b}$ and model predictions $\hat{\mathbf{y}}_{k,b}$. These vectors will be aggregated and evaluated with a scoring function, $s(\mathbf{y}, \hat{\mathbf{y}})$. The two scoring functions considered are auROC and AP*.

The empirical in-distribution performance, $P_{\text{in-dist},s(\cdot),f}(D)$, of the predictive model $f$ on the given dataset $D$ using scoring function $s(\cdot)$ is:

$$\mathbf{y}_{\text{micro-avg},b} = \texttt{concat}(\mathbf{y}_{k,b} \;\forall\; k) \tag{8}$$

$$\hat{\mathbf{y}}_{\text{micro-avg},b} = \texttt{concat}(\hat{\mathbf{y}}_{k,b} \;\forall\; k) \tag{9}$$

$$P_{\text{in-dist},s(\cdot),f}(D) = \frac{1}{B} \sum_b s(\mathbf{y}_{\text{micro-avg},b}, \hat{\mathbf{y}}_{\text{micro-avg},b}). \tag{10}$$

*Out-of-distribution Analysis Methodology:* To evaluate how well a model generalizes across datasets, we define a training dataset $D_{\text{train}}$ and a test dataset $D_{\text{test}}$. Training and evaluation are repeated $B$ times independently. Re-using notation from the previous section (and redefining a new number of bootstrap iterations $B$ and $b \in \{1, \ldots, B\}$), this bootstrapped out-of-distribution evaluation on the test set gives scalar numbers of the form $s(\mathbf{y}_{\text{test},b}, \hat{\mathbf{y}}_{\text{test},b})$ where $\mathbf{y}_{\text{test},b}$ contains the ground truth test dataset annotations, and $\hat{\mathbf{y}}_{\text{test},b}$ contains the trained model's test set predictions.

We compare the model's performance to the performance of a separately trained in-distribution baseline model by computing a ratio. This ratio is averaged over B bootstrap iterations:

$$P_{\text{out-of-dist},s(\cdot),D_{\text{train}},f}(D_{\text{test}}) = \frac{1}{B} \sum_b s(\mathbf{y}_{\text{test},b}, \hat{\mathbf{y}}_{\text{test},b}) \tag{11}$$

$$P_{\text{ood-ratio},D_{\text{train}},D_{\text{test}},f} = \frac{\text{out-of-distribution score}}{\text{in-distribution score}} \tag{12}$$

$$= \frac{P_{\text{out-of-dist},s(\cdot),D_{\text{train}},f}(D_{\text{test}})}{P_{\text{in-dist},s(\cdot),f}(D_{\text{test}})}, \tag{13}$$

where we have a given scoring function $s(\cdot)$ implementing the auROC or AP*, a training dataset $D_{\text{train}}$, an evaluation dataset

TABLE II
DATASET SUMMARY

| Dataset | Domain | Mode | Subjects | Samples | PH Samples |
|---|---|---|---|---|---|
| H-PCG [15] | Human | PCG | 42 | 42 | 29 (69%) |
| P-Both [28] | Porcine | PCG | 10 | 32 | 16 (50%) |
| | Porcine | SCG | 10 | 93 | 42 (45%) |
| H-SCG [29] | Human | SCG | 73 | 82 | 63 (76%) |

$D_{\text{test}}$, a predictive model $f$, and an in-distribution evaluation function $P_{in-dist,s(\cdot),f}$ from (10).

Note that the denominator of (11) is effectively a baseline against which the out-of-distribution score is compared. The baseline can be chosen arbitrarily, or in a way that aids interpretation. For instance, for model selection, the denominator can show how much performance is lost when compared to the best model. More specifically, given a set of models $\mathcal{G}$, where each model $g \in \mathcal{G}$ is trained on $D_{\text{train}}$ and evaluated on $D_{\text{test}}$, we define the denominator as the maximum in-distribution score of the test dataset across all models. This variation of the score is used in our experiments, and is shown below:

$$P_{\text{ood-ratio},D_{\text{train}},f,G} = \frac{P_{\text{out-of-dist},s(\cdot),D_{\text{train}},f}(D_{\text{test}})}{\max_{g\in\mathcal{G}} P_{\text{in-dist},s(\cdot),g}(D_{\text{test}})} \tag{14}$$

Given the predictive model $f$ and scoring function, the (14) gives one measurement for each pair of datasets. To give a general sense of performance comparable across all datasets, we propose to compute the median over the pairs of datasets in order to have a single number for each model and scoring function. We define a set of pairs of train and test datasets as $\mathcal{D}$, and compute the median score, where $m(\cdot)$ is the median function:

$$P_{\text{ood-avg},s(\cdot),f,G} = m(\{P_{\text{ood-ratio},D_{\text{train}},f,G} \| (D_{\text{train}}, D_{\text{test}}) \in \mathcal{D}\}). \tag{15}$$

### C. Datasets

We evaluate three different datasets. The datasets vary in recording modality, animal species and hospital location. The modalities considered include phonocardiography (PCG), and seismocardiography (SCG); PCG uses microphones while SCG uses accelerometers. Two datasets contain data from humans and one dataset is from pigs. All datasets are from different hospitals. All three datasets record subjects undergoing right heart catheterization. The datasets are described in the following paragraphs and summarized in Table II.

***1) Dataset H-PCG:*** Human PCG data was acquired from 42 subjects at Centro Hospitalar Universitário do Porto, Portugal. 29 subjects have PH and 13 subjects do not have PH. The dataset was previously introduced with the name HSA [15] and it is not publicly available.

For each subject, ground truth pulmonary artery pressure was obtained from a right heart catheterization (Swan-Ganz catheter), and an accompanying five-minute PCG heart sound recording was collected. The recording was obtained in a quiet clinical setting with the patient supine and at rest. Auscultation was performed over the second left intercostal space using a custom cable stethoscope connected to a Rugloop Waves system. Heart sounds were recorded at a sample rate of 8 kHz and their amplitudes were quantized with 16-bit resolution. We sub-sample the signal to 1 kHz. The 200 ms time window vectors for each recording were obtained by the CNN-based method of Renna et al. [30]. The subjects in the dataset are labeled as PH positive if have Mean Pulmonary Arterial Pressure (MPAP) above 25 mmHg, or Pulmonary Arterial Systolic Pressure (PASP) above 30 mmHg.

***2) Dataset P-Both:*** Porcine PCG data (P-PCG) and SCG data (P-SCG) were acquired at Aalborg University Hospital, Aalborg, Denmark [28]. The pigs were sedated, ventilated, and subject to catheterism both in the aorta and the right ventricle using Swan-Ganz catheters. The SCG acquisition was carried out using an iWorx commercial system equipped with two triaxial accelerometers, located respectively over the fourth intercostal space next to the sternum and over the lower border of the sternum. The PCG acquisition was carried out using a multi-channel wearable system designed at Politecnico di Torino, which embeds 48 electret condenser microphones with a 12-mm spatial resolution [31]. A simultaneous ECG was collected by both instruments. The sampling frequency while recording was set to 5 kHz for SCG and to 1 kHz for PCG, and then we downsampled the SCG signal to 1 kHz using nearest-neighbor interpolation. A PH condition was triggered in the animal by subjecting it to either nitrogen asphyxiation (to cause hypoxemia) or carbon dioxide asphyxiation (to cause hypercapnia). The experiment was reversible, and therefore when the trigger was removed, the animal reached a baseline condition again and could be subjected to multiple experiments. A total of 59 experiments were carried out on ten pigs. The dataset for this study was created by considering, for each recording, two one-minute segments: one at the beginning of the recording, labeled as "no PH", and one at the peak of the effect of the trigger, labeled as "PH". Poor-quality recordings were manually discarded. There were 67 "no PH" and 58 "PH" samples. For the segmentation pre-processing steps, we applied a band-pass of 20 Hz to 200 Hz using a second-order IIR Butterworth filter, then the signals were segmented into heartbeats using the R-wave ECG, and then a 200-millisecond S2 segment was extracted from each heartbeat using time thresholding and peak detection. The full dataset P-Both is a union of the P-SCG and P-PCG datasets, where P-SCG and P-PCG share the same physiological basis but have different modalities. The P-Both dataset is analyzed only as a training dataset in the context of out-of-distribution experiments with human test datasets, and it enables the analysis of whether training on only PCG data, only SCG data, or a mixture of both modalities from the same set of subjects can improve generalization performance on humans.

The dataset is not publicly available.

***3) Dataset H-SCG:*** The dataset was acquired at the catheterization laboratory at the University of California and is publicly available on PhysioNet [29] with the name SCG-RHC. The dataset includes 83 recordings of 72 patients referred for the hemodynamic assessment of their heart failure status. While the dataset includes data from a pharmacological experiment, we use only data from the end of the 10 minute patient resting period that occurred before any pharmacological intervention. ECG and

triaxial SCG were simultaneously recorded using a wearable patch located over the sternum. The sampling frequency is 500 Hz. A subject was labeled as PH positive if the PASP was greater than 30 mmHg or if the MPAP was greater than 25 mmHg. The segmentation pre-processing obtained 200 ms windows of each recorded S2 by the method used for the P-Both dataset, and the S2 segments were aligned in time by selecting a lag index that maximizes the cross-correlation between the homomorphic envelogram of the segment and the homomorphic envelogram of the first S2 segment of the recording.

### D. Hyperparameters and Competing Methods

A total of four methods are evaluated: PH-ELM, DenseNet121, Scattering-SVM, and STFT-SVM. The reasoning for the choice of models and description of their construction is presented below. The DenseNet121 deep network [32] was chosen for two reasons: a) our previous work found DenseNet121 to outperform the in-distribution performance of several other models on the H-PCG dataset [15]; and b) the PH-ELM model's fixed-weight convolutional feature generator is similar to a deep convolutional network. The Scattering-SVM consists of a wavelet scattering feature generator and support vector machine (SVM). The wavelet scattering component was chosen due to the similarity of scattering to the PH-ELM's feature generation, and the SVM was chosen because literature comparing SVM to ELM has found that SVM can outperform ELM models on small datasets [33]. The STFT-SVM model, previously introduced in [15], utilizes a short-time Fourier transform for feature generation followed by an SVM. It was chosen because it is a resource-efficient method that can perform well.

*Hyperparameters for DenseNet121:* The models utilizing the DenseNet121 architecture were randomly initialized (they were not pre-trained) and optimized with the AdamW optimizer [34] with learning rate of 1e-5 and weight decay of either 1-e5 (P-Both) or 1e-6 (H-PCG and H-SCG). Because the model cannot handle images with less than 61 rows or columns, the input to DenseNet121 was zero padded to a uniform size of 454 rows (H-PCG) or 157 rows (P-Both, P-SCG, and P-PCG). Padding on H-SCG required additional treatment, as some patients have too few rows, and others have too many rows. Each input image recording was zero padded to 400 rows, and recordings with too many rows were cropped to 400 contiguous rows. When cropping, the crop region was randomly chosen during training in order to use the entire dataset, and fixed to the last 400 rows of the recording (but before the H-SCG chemical experiment began) during testing.

*Hyperparameters for Scattering-SVM:* Since the SVM requires a fixed-size input that is the same in both training and testing, the input to Scattering-SVM was also zero padded. We adopt the padding approach and random cropping approach used for DenseNet121, but fix all datasets to 400 rows. Moreover, the columns of each padded (or cropped) image were normalized to variance 1 because we found empirically that this improved performance. The scattering layer used 2 orders, wavelet scale $J = 2, L = 8$ angles, the Morlet wavelet family, and the standard implementation from the Kymatio library [35]. After scattering, each input was reshaped into a row vector. For training, all inputs are stacked into a matrix and each feature column was normalized to zero mean and unit variance. This matrix was passed to the support vector regression from SciKit Learn [26] initialized with $C = 1.0$, RBF kernel, and $\gamma = \frac{1}{C(X)\sigma^2(X)}$ (also known as "scale" in SciKit Learn) where $C(X)$ is the number of training set columns ($C(X) = 200$ in all datasets) and $\sigma^2$ is the variance of the training dataset.

*Hyperparameters for STFT-SVM:* The padding methodology was identical to Scattering-SVM.[3] The STFT was performed by computing a spectrogram from the PyTorch Audio library [36] for each row (each heartbeat), using FFT of size 64 (giving 33 frequency bins), hop length 2, and power 1 for the magnitude spectrum. The spectra from all heartbeats were aggregated by computing the 98% and 100% quantiles across rows, reducing the size of each subject's recording from (H, 200) to (33, 101), where rows are frequency bins, and columns are time. The columns of this matrix were normalized to unit variance and then flattened into a row vector. The SVM (from SciKit Learn) used $C = 1$, RBF kernel, and $\gamma = \frac{1}{C(X)}$ (also known as "auto" in SciKit learn). This model was previously introduced in [15].

*Hyperparameters for PH-ELM:* The PH-ELM's convolutional feature generator used $O = 1000$ output channels, kernel size $5 \times 5$, dilation $2 \times 2$, $G = 10$ groups, and 20 principal components. The linear regression used $\lambda = 0.1$. Padding was not necessary for this model because the feature generator standardizes all inputs to the same size.

## IV. Results

We evaluate the four PH detection models and three datasets described in Section III-C. Sections IV-A and IV-B respectively present the observed in-distribution and out-of-distribution prediction performances, Section IV-C presents the computational performance, and Section IV-D examines the effect of varying heartbeats per subject.

### A. In-Distribution Prediction Performance

Table III is useful primarily to give a sense of the overall best prediction performance a model might expect to achieve on a given dataset. The calculation for auROC and AP* scores is defined by (10). The DenseNet121, followed by PH-ELM, tends to have the highest prediction performances in both AP* and auROC across the three datasets. The PH-ELM model has the lowest standard deviation of all models on the H-PCG and H-SCG. Moreover, we can observe that the datasets can be ranked based on performance in this order: H-PCG (highest), P-Both, and H-SCG (lowest). Possible reasons for H-SCG's low performance might be related to the placement of the SCG sensor on the sternum, or to the dataset's 500 Hz sampling rate. The other datasets utilized the second and fifth intercostal spaces, respectively, and they were sampled at 1000 Hz. We do not make comparisons of PCG and SCG signals for PH detection based on this in-distribution analysis.

[3]The computational footprint tests in Table VI do not use padding, as padding is not necessary for this model and would increase the resource footprint.

TABLE III
IN-DISTRIBUTION PREDICTION PERFORMANCE – VIA (10)

| Dataset | Model | auROC | AP* | $\sigma$(auROC) | $\sigma$(AP*) |
|---|---|---|---|---|---|
| H-PCG | DenseNet121 | **0.932** | **0.728** | 0.029 | 0.038 |
| H-PCG | PH-ELM | 0.922 | 0.726 | 0.012 | 0.015 |
| H-PCG | STFT-SVM | 0.891 | 0.695 | 0.023 | 0.024 |
| H-PCG | Scattering-SVM | 0.827 | 0.573 | 0.019 | 0.047 |
| H-SCG | DenseNet121 | 0.547 | 0.274 | 0.062 | 0.038 |
| H-SCG | PH-ELM | **0.650** | **0.335** | 0.024 | 0.021 |
| H-SCG | STFT-SVM | 0.541 | 0.271 | 0.032 | 0.012 |
| H-SCG | Scattering-SVM | 0.537 | 0.233 | 0.039 | 0.027 |
| P-Both | DenseNet121 | **0.874** | **0.820** | 0.019 | 0.024 |
| P-Both | PH-ELM | 0.816 | 0.765 | 0.001 | 0.002 |
| P-Both | STFT-SVM | 0.656 | 0.622 | 0.000 | 0.000 |
| P-Both | Scattering-SVM | 0.694 | 0.574 | 0.000 | 0.000 |
| P-PCG | DenseNet121 | 0.924 | **0.916** | 0.023 | 0.024 |
| P-PCG | PH-ELM | 0.894 | 0.872 | 0.007 | 0.019 |
| P-PCG | STFT-SVM | 0.720 | 0.741 | 0.000 | 0.000 |
| P-PCG | Scattering-SVM | **0.934** | 0.903 | 0.000 | 0.000 |
| P-SCG | DenseNet121 | **0.860** | **0.784** | 0.024 | 0.045 |
| P-SCG | PH-ELM | 0.784 | 0.732 | 0.002 | 0.004 |
| P-SCG | STFT-SVM | 0.609 | 0.509 | 0.000 | 0.001 |
| P-SCG | Scattering-SVM | 0.697 | 0.546 | 0.000 | 0.000 |

Each row shows the average performance of 12 independent bootstrap iterations with cross-validation. P-Both had leave-one-pig-out CV (10 pigs), H-PCG had 10-fold CV, and H-SCG had 10-fold grouped CV to ensure that each subject's data was entirely in either the train or the validation set.

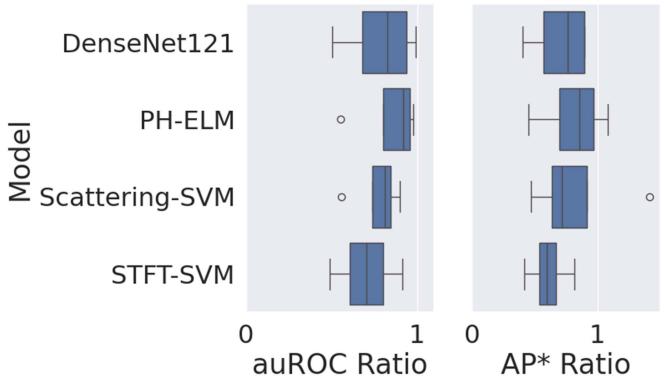


Fig. 3. **Model Selection:** All models have similar prediction performance, and the PH-ELM has the highest median out-of-distribution performance when evaluating H-PCG or H-SCG test sets with any of the training sets {P-Both, H-PCG or H-SCG}, with median auROC ratio and median AP* ratio of +0.09 (in both cases) over DenseNet121. Each boxplot shows 0%, 25%, 50%, 75%, and 100% percentiles of 12 bootstrap iterations. This figure summarizes rows from Tables IV and V.

## B. Out-of-Distribution Prediction Performance

Out-of-distribution testing shows the prediction performance when training a model on one dataset and evaluating it on a different dataset.

*PH detection with SCG and PCG datasets:* Table IV shows PH detection performance when training with PCG signals to evaluate SCG signals, and vice versa, in either humans (comparing H-SCG to H-PCG) or pigs (comparing P-SCG to P-PCG). While the human datasets are independent, the porcine datasets are generated from the same ten pigs. To ensure the train and test sets do not share the same physiological basis, we utilized a leave-one-pig-out cross-validation, with an additional constraint that the PCG and SCG data are disjoint across train and test sets (i.e., either $D_{\text{train}} \subset$ P-PCG and $D_{\text{test}} \subset$ P-SCG, or $D_{\text{train}} \subset$ P-SCG and $D_{\text{test}} \subset$ P-PCG), and the reported results are obtained from the *in-distribution* (10). We identify the porcine experiments as out-of-distribution because we analyze generalization across modalities (PCG and SCG), but the non-independent nature of the datasets requires cross-validation and the in-distribution evaluation equation.

Regarding model selection, the PH-ELM model has the highest scores (in bold) for all pairs of datasets except when $(D_{\text{train}}, D_{\text{test}})$ is (H-SCG, H-PCG), in which case the model ranks second to Scattering-SVM (with auROC of 0.55 versus 0.83). DenseNet121 and STFT-SVM both become no better than random from H-SCG to H-PCG (auROC of $0.5 \pm .01$). Moreover, of the three instances in which a model gives *higher* out-of-distribution performance than in-distribution performance, two of these occur with the PH-ELM, and these are shown by an AP* Ratio or auROC Ratio greater than one. The results suggest that the PH-ELM generalizes well across PCG and SCG signals with little loss in prediction performance. In summary, the PH-ELM shows the best out-of-distribution generalization performance overall.

The table also offers some insight into dataset selection. Since the H-SCG dataset has the overall lowest in-distribution scores in Table III, it may not be a high-quality training dataset for PH detection, since three of the four models (DenseNet121, PH-ELM, and STFT-SVM) give performances that are effectively no better than random when training H-SCG to evaluate H-PCG. The fact that Scattering-SVM gave good performance on H-SCG, though, means the dataset does retain information about PH. On the other hand, P-SCG $\rightarrow$ P-PCG shows comparably good generalization with three of the four models. Both results can be interpreted to suggest that SCG signals can provide a useful training dataset for PH detection in PCG signals. Finally, the evaluation PCG $\rightarrow$ SCG, using either human or porcine datasets, gives useful models. In summary, the results show that both PCG and SCG data can be useful as training sets.

*PH detection from pigs to humans:* Table V shows that porcine heart sounds can provide useful training data to predict PH in human heart sounds. The performance from humans to pigs was excluded from the table because we assume that the generalization from humans to pigs is not relevant. The generalization to pigs has auROC values less than 0.6 in nearly all cases, possibly due to the way the P-Both dataset defines PH as a part of its experiment design. From a model selection point of view, the PH-ELM is best for generalizing to H-PCG, and for generalization to H-SCG, it is unclear which model is preferable.

Regarding dataset selection, we interpret the results to suggest that more data can improve performance because using the full P-Both training dataset gives higher performance than either P-SCG or P-PCG alone. This result may suggest that combining modalities (both PCG and SCG) may improve performance.

*The PH-ELM generalizes the best:* Fig. 3 summarizes the prediction performance of the evaluated models by showing box plots of the min, 25%, median, 75%, and maximum prediction

TABLE IV
SCG ↔ PCG, OUT-OF-DISTRIBUTION PREDICTION PERFORMANCE

| $D_{\text{train}}$ | $D_{\text{test}}$ | Model | AP* Ratio | auROC Ratio | AP* | auROC | $\sigma$(AP*) | $\sigma$(auROC) |
|---|---|---|---|---|---|---|---|---|
| H-SCG | H-PCG | DenseNet121 | 0.401 | 0.506 | 0.292 | 0.471 | 0.042 | 0.068 |
| H-SCG | H-PCG | PH-ELM | 0.452 | 0.553 | 0.329 | 0.515 | 0.020 | 0.020 |
| H-SCG | H-PCG | STFT-SVM | 0.418 | 0.491 | 0.304 | 0.458 | 0.036 | 0.002 |
| H-SCG | H-PCG | Scattering-SVM | **0.746** | **0.828** | **0.543** | **0.771** | 0.068 | 0.094 |
| H-PCG | H-SCG | DenseNet121 | 0.897 | 0.917 | 0.300 | 0.596 | 0.027 | 0.028 |
| H-PCG | H-SCG | PH-ELM | **1.082** | **0.978** | **0.362** | **0.636** | 0.004 | 0.005 |
| H-PCG | H-SCG | STFT-SVM | 0.819 | 0.912 | 0.274 | 0.593 | 0.000 | 0.000 |
| H-PCG | H-SCG | Scattering-SVM | 0.689 | 0.897 | 0.231 | 0.583 | 0.000 | 0.000 |
| P-SCG | P-PCG | DenseNet121 | **0.817** | **0.944** | **0.748** | **0.872** | 0.017 | 0.007 |
| P-SCG | P-PCG | PH-ELM | 0.650 | 0.821 | 0.595 | 0.759 | 0.005 | 0.003 |
| P-SCG | P-PCG | STFT-SVM | 0.491 | 0.585 | 0.450 | 0.540 | 0.001 | 0.001 |
| P-SCG | P-PCG | Scattering-SVM | 0.545 | 0.636 | 0.499 | 0.588 | 0.009 | 0.015 |
| P-PCG | P-SCG | DenseNet121 | 1.136 | 1.057 | 0.891 | 0.909 | 0.047 | 0.041 |
| P-PCG | P-SCG | PH-ELM | **1.213** | **1.125** | **0.951** | **0.967** | 0.030 | 0.021 |
| P-PCG | P-SCG | STFT-SVM | 0.509 | 0.530 | 0.399 | 0.456 | 0.000 | 0.000 |
| P-PCG | P-SCG | Scattering-SVM | 0.694 | 0.619 | 0.544 | 0.532 | 0.023 | 0.006 |

Each row shows the average performance of 12 independent bootstrap iterations for the given training dataset, test dataset, and model.

TABLE V
PIG → HUMAN, OUT-OF-DISTRIBUTION PREDICTION PERFORMANCE

| $D_{\text{train}}$ | $D_{\text{test}}$ | Model | AP* Ratio | auROC Ratio | AP* | auROC | $\sigma$(AP*) | $\sigma$(auROC) |
|---|---|---|---|---|---|---|---|---|
| P-Both | H-PCG | DenseNet121 | 0.630 | 0.735 | 0.458 | 0.685 | 0.059 | 0.052 |
| P-Both | H-PCG | PH-ELM | **0.781** | **0.883** | **0.568** | **0.822** | 0.013 | 0.007 |
| P-Both | H-PCG | STFT-SVM | 0.576 | 0.642 | 0.419 | 0.598 | 0.001 | 0.001 |
| P-Both | H-PCG | Scattering-SVM | 0.468 | 0.555 | 0.341 | 0.517 | 0.000 | 0.000 |
| P-PCG | H-PCG | DenseNet121 | 0.464 | 0.642 | 0.338 | 0.598 | 0.031 | 0.030 |
| P-PCG | H-PCG | PH-ELM | **0.544** | **0.682** | **0.396** | **0.635** | 0.006 | 0.006 |
| P-PCG | H-PCG | STFT-SVM | 0.430 | 0.473 | 0.313 | 0.440 | 0.000 | 0.000 |
| P-PCG | H-PCG | Scattering-SVM | 0.000 | 0.518 | 0.000 | 0.483 | 0.000 | 0.000 |
| P-SCG | H-PCG | DenseNet121 | **0.519** | 0.649 | **0.378** | 0.605 | 0.069 | 0.080 |
| P-SCG | H-PCG | PH-ELM | 0.517 | **0.684** | 0.376 | **0.637** | 0.032 | 0.034 |
| P-SCG | H-PCG | STFT-SVM | 0.490 | 0.640 | 0.357 | 0.596 | 0.001 | 0.001 |
| P-SCG | H-PCG | Scattering-SVM | 0.468 | 0.555 | 0.341 | 0.517 | 0.000 | 0.000 |
| P-Both | H-SCG | DenseNet121 | 0.897 | **0.991** | 0.300 | **0.645** | 0.030 | 0.026 |
| P-Both | H-SCG | PH-ELM | 0.928 | 0.954 | 0.311 | 0.620 | 0.020 | 0.004 |
| P-Both | H-SCG | STFT-SVM | 0.620 | 0.763 | 0.207 | 0.496 | 0.001 | 0.001 |
| P-Both | H-SCG | Scattering-SVM | **1.415** | 0.796 | **0.474** | 0.518 | 0.000 | 0.000 |
| P-PCG | H-SCG | DenseNet121 | 0.821 | 0.896 | 0.275 | 0.582 | 0.019 | 0.018 |
| P-PCG | H-SCG | PH-ELM | 0.822 | **0.927** | 0.275 | **0.603** | 0.001 | 0.002 |
| P-PCG | H-SCG | STFT-SVM | 0.618 | 0.645 | 0.207 | 0.419 | 0.000 | 0.000 |
| P-PCG | H-SCG | Scattering-SVM | **1.415** | 0.796 | **0.474** | 0.518 | 0.000 | 0.000 |
| P-SCG | H-SCG | DenseNet121 | 0.815 | **0.916** | 0.273 | **0.596** | 0.043 | 0.050 |
| P-SCG | H-SCG | PH-ELM | 0.813 | 0.885 | 0.272 | 0.575 | 0.013 | 0.014 |
| P-SCG | H-SCG | STFT-SVM | 0.792 | 0.849 | 0.265 | 0.552 | 0.001 | 0.001 |
| P-SCG | H-SCG | Scattering-SVM | **1.415** | 0.796 | **0.474** | 0.518 | 0.000 | 0.000 |

Each row shows the average performance of 12 independent bootstrap iterations for the given training dataset, test dataset, and model.

performances over selected rows of Tables IV and V (the rows corresponding to the P-SCG and P-PCG datasets are excluded since they are covered by P-Both). The median is derived via (15). The results show that the PH-ELM model has the highest median performance by +0.09 in both auROC Ratio and AP* Ratio over the closest model, DenseNet121. Indeed, the median auROC scores are 0.884 (PH-ELM), 0.815 (DenseNet121), 0.796 (Scattering-SVM), and 0.643 (STFT-SVM). In general, the PH-ELM model does the best job of preserving generalization performance that could be obtained by in-distribution analysis.

### C. Computational Performance

Table VI shows the computational footprint of the four models. Two columns in the table count the total number of learned and fixed (not learned) parameters for each model. To count parameters in an SVM model, we count the number of coordinates across all support vectors of the SVM trained on the entire H-PCG dataset. Since the number of support vectors depends on dataset size, this measurement can vary, but there were 42 support vectors after training on H-PCG (one per subject in the dataset) for both STFT-SVM and Scattering-SVM. The values reported in columns Time to Train and Power to

TABLE VI
COMPUTATIONAL RESOURCE FOOTPRINT

| Model | Learned Parameters | Not Learned Parameters | Time to Train[1] (s) | Power to Train[1] (W) | Inference Time per Subject[2] (s) |
|---|---|---|---|---|---|
| DenseNet121[3] | 6.95e6 | 0.084e6 | 318.8 ± 1.16 | 89,481 ± 92 | 0.062 ± 0.044 |
| PH-ELM | **0.02e6** | 0.03e6 | **8.80** ± 0.04 | **168** ± 17 | 0.0014 ± 0.0014 |
| Scattering-SVM | 17.8e6** | 1.46e6 | 19.74 ± 0.14 | 2020 ± 36 | 0.0356 ± 0.0011 |
| STFT-SVM | 0.14e6** | **64** | 12.15 ± 0.05 | 370 ± 8 | **0.00016** ± 0.00002 |

** The parameter count in SVM models depends on the number of support vectors, which depends on the training set and hyperparameter constraint $C$, and is therefore variable. The full H-PCG dataset was utilized to give parameter count estimates, and the SVM models learned 42 support vectors after training. The Scattering parts of Scattering-SVM contribute non-learned parameters.
1. Time to Train and Power to Train computed on a server with NVIDIA RTX 2080 GPU and AMD Ryzen Threadripper 2920X 12-Core Processor. Power is obtained by sampling the CPU and GPU watt usage every second, summing over the number of seconds of training time, and averaging across 3 runs.
2. Inference Time per Subject computed on a notebook with 13th Gen Intel(R) Core(TM) i7-13700H CPU and NVIDIA RTX 4060 GPU. SVM models are evaluated on the CPU, though the wavelet scattering for ScatteringSVM is computed on the GPU. Time may vary widely from system to system. Results average the performance across all samples in the H-PCG dataset.
3. DenseNet121 evaluated for 500 epochs. Training could be completed in as few as 150 epochs, at negligible performance loss, for most datasets.

Train each report the average and standard deviation of three measurements. Each measurement was performed on the same dedicated Linux server, running no other jobs, with an NVIDIA RTX 2080 GPU and AMD Ryzen Threadripper 2920X 12-Core CPU. The time estimate for each measurement was obtained by training on H-PCG and evaluating on H-SCG (we assume the added overhead of time spent evaluating H-SCG is negligible). Independently of the timing estimates, each wattage estimate was obtained by sampling the Linux operating system (at location `/sys/class/powercap/*/energy_uj`) for CPU power usage and by sampling the NVIDIA graphics card (via the Linux tool `nvidia-smi`) for power usage. The watt numbers were recorded once per second during the period of time when the model was training (not evaluating) and then summed together. The Inference Time per Subject was computed by averaging the elapsed real time spent to generate a prediction of a single sample (i.e. batch size of one sample), using a notebook computer, Lenovo P1, with a 13th Gen Intel(R) Core(TM) i7-13700H CPU and NVIDIA RTX 4060 GPU. All variations in the table show one standard deviation.

The PH-ELM trains most quickly, requires the least amount of power to train, has the smallest number of learned parameters, the smallest number of parameters (0.047e6) overall, and is fast for inference. In contrast, the DenseNet121, which is the most competitive model with regards to classification performance, has over 300× more learned parameters, takes 36× longer to train on a decent quality GPU, and utilizes over 500× more watts of power to train. Note also that the PH-ELM is non-iterative. As the dataset size increases, the time and power required to train a deep network (or SVM model) will increase significantly due to the iterative nature of the algorithms, whereas the PH-ELM is bounded by the time needed to compute features from a single convolution layer. Finally, the STFT-SVM model is notable for extremely fast inference speed and nearly zero not learned parameters, but it ranks last by a large margin with regards to the the median prediction performance shown in Fig. 3.

### D. Varying Number of Heartbeats

This section analyzes if there is a minimum number of heartbeats per subject necessary for PH detection. For instance, it would be beneficial to know if a clinician only needs a short heart sound recording for PH detection. Moreover, we can also ask whether a longer recording improves the quality of the training data.

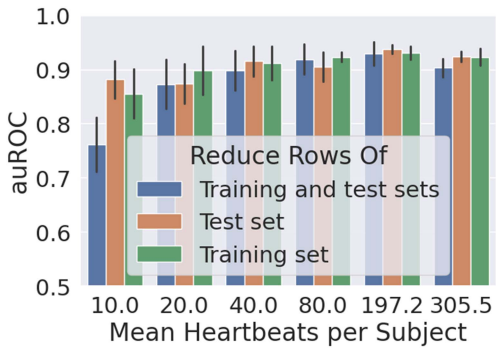


Fig. 4. **Insensitive to a varying number of Heartbeats:** In-distribution prediction performance with the H-PCG dataset and PH-ELM model is almost unchanged when either the test data or training data has as few as 10 heartbeats. Each bar shows the average and one standard deviation of 12 independently trained PH detectors.

Fig. 4 shows the effect of reducing the number of heartbeats per subject in either the training set, validation set, or both. This test is performed on the H-PCG dataset via cross-validation with the PH-ELM model, and the figure reports the in-distribution auROC via (10) and 12 bootstrap iterations. Each bar in the barplot reports the average and one standard deviation.

We observe that the PH-ELM model is somewhat insensitive to the varying number of heartbeats. While more heartbeats per subject may slightly increase the prediction performance, 40 heartbeats may offer an acceptable trade-off between recording time and performance. We interpret the figure to imply that the mean number of heartbeats in the training set is slightly more important than in the test set, since at 10 mean heartbeats per subject, the reported auROC drops more when reducing the training set (green bar) than the test set (orange bar).

## V. DISCUSSION

**The automated detection of Pulmonary Hypertension from heart sound analysis is a viable and robust technology that generalizes across domains and modalities.** The results

of our study show strong PH detection performance to human test data across all available combinations of three datasets. We applied a rigorous evaluation including two different modalities (PCG and SCG), two animals (humans and pigs), and in-distribution and out-of-distribution evaluation. The model with the best prediction performance, PH-ELM, is also computationally lightweight and well-suited to deployment on edge devices such as mobile phones or small portable PH detection machines. Our study offers evidence that a heart sound PH detector could be recognized in clinical guidelines as a useful and reliable technology for the detection of PH. A limitation of the datasets considered is that all subjects underwent the right heart catheterization, which means the presentation of PH is sufficiently advanced to warrant an invasive and expensive procedure. The results of our work justify the need for studies of *early* PH detection, such as by following the long-term outcomes of presumably healthy individuals who are regularly subjected to PH screenings alongside existing technologies like ECHO, EKG, and blood tests.

### A. In-Distribution and Out-of-Distribution Testing

**Out-of-distribution testing offers benefits not available from in-distribution testing:** The analysis of related work in Section II and Table I shows that none of the works surveyed have evaluated the empirical out-of-distribution performance with a separate dataset, possibly due to the challenge of obtaining PH datasets. Our present work therefore contributes improved evaluation techniques by utilizing both in-distribution and out-of-distribution analyses.

**In-distribution performance reported on any single dataset is not representative of expected out-of-distribution results.** While Table III reports an in-distribution standard deviation, the reported standard deviation can be made arbitrarily smaller by increasing the number of bootstrap iterations, a phenomenon explained by the Central Limit Theorem (CLT) [37]. This variability for a given dataset is not representative of the variability of the expected out-of-distribution performance when the dataset is used as a training or test set. Moreover, if the expected value of the in-distribution auROC was predictive of the out-of-distribution auROC, we would see an auROC ratio close to one, and we do not observe that in Tables IV and V; the same logic applies to AP*. In Table V, the expected test set auROC of the PH-ELM model is 0.82, 0.64, 0.64, 0.62, 0.60, and 0.58 (each also with small standard variation due to the CLT and twelve bootstrap iterations). Excluding the value 0.82, there is a relatively small variation in these results. The 0.82 auROC value appears when the dataset P-Both is the training set and H-PCG is the test set. The DenseNet121 follows this trend, with its highest performance of 0.69 auROC when (P-Both, H-PCG) are the (training, test) sets. Since P-Both combines SCG and PCG data, this out-of-distribution result offers preliminary evidence that training with multiple modalities (PCG and SCG data) may improve performance on PCG test data. Comparing these reported results to the in-distribution results in Table III, we can observe that no single in-distribution number reliably predicts the out-of-distribution performances.

**In-distribution results can mislead model selection in a possible real-world setting.** A realistic setting could arise where we have P-Both training data, and we wish to train and deploy a model to the hospital where H-PCG data is generated. Based on the in-distribution results with P-Both, we would choose DenseNet121. However, the corresponding out-of-distribution experiments (P-Both, H-PCG) in the first two rows of Table V show that the PH-ELM is preferable over DenseNet121 by 0.14 auROC. Model selection based on the reported in-distribution results can therefore be misleading.

**The machine learning literature justifies the benefits of OOD testing:** A recent work claims that in-distribution and out-of-distribution performance can be inversely correlated, and therefore that "studies on OOD generalization that use ID performance for model selection (a common recommended practice) will necessarily miss the best-performing models, making these studies blind to a whole range of phenomena" [38], where ID means in-distribution. Similar works describe underspecification as a phenomenon of large variations in test time performance from a set of models that have equally good in-distribution performance [39]. Spurious correlation, as surveyed in [40], can also explain some kinds of test set variability. The literature therefore provides ample evidence that in-distribution and out-of-distribution performances can be completely different, and even inversely correlated. This literature shows that using cross-validation results, which is an in-distribution test, to perform model selection for application in an out-of-distribution setting, is not necessarily justified or reliable.

### B. PH-ELM Model Design

The ELM architecture was originally designed for prediction tasks on tabular data [41], especially small datasets [42], using random and fixed random projection matrix followed by a nonlinear operator to generate features, and there are many variations of the basic architecture [43]. The application of ELM to images involves converting the 2-D or 3-D image data into a 1-D vector [44]. Previous works have trained a CNN using backpropagation and then utilized its outputs as a feature extractor for an ELM model, such as in the analysis of retinal fundus images for diabetic retinopathy [45], and to detect QRS complex in heart electrocardiograms [46]. In contrast, our approach does not train the CNN or use an iterative optimization algorithm. Similarly to our work, a fixed-weight convolutional layer followed by feature pooling was applied to images, where the convolutional kernel weights were initialized with Gaussian random parameters and then orthogonalized using SVD[4] [47]. Some works have considered SVD compression and ELM; singular values were utilized as the input to an ELM classifier [48], and another work shows that using PCA to compress the data just before passing it to an ELM classifier increases computational efficiency [49]. Our approach is distinct from these related works. We adopt a variation of the ELM architecture, where (a) our parallelized CNN implementation with an adaptive pooling function replaces

[4]In [47], SVD was for parameter initialization, not for compression of the generated convolutional features.

the ELM's hidden layer, (b) we apply a PCA projection matrix to compress the CNN feature representation, and (c) we assemble the training data into a matrix and compute a pseudo-inverse like a standard ELM procure would.

Another machine learning pipeline that is similar to our ELM-based approach is the wavelet scattering network [50], which has been described as a convolutional network [51] with a fixed-weight architecture containing one to three convolutional structures, each followed by a non-linear operator. Wavelet scattering outputs a data structure that can be compressed, such as with PCA, and then used to train or evaluate a linear model [51]. Similar approaches based on wavelet packet compression [52] also use a single convolution layer and demonstrate no loss in prediction performance when parameters from the convolutional structure are removed or obscured. Convolution layers are linear functions by design, and they therefore fit into the ELM framework. These works therefore justify the PH-ELM's adoption of a convolutional feature-generating structure with fixed-weight parameters, and they also justify our comparative evaluations of the Scattering-SVM model.

Few works have previously applied the ELM architecture to heart sound analysis [53], [54], [55], [56]. To the best of our knowledge, there are no existing published works that apply ELM to SCG data. The work of Liu et al. [53] utilized a standard ELM architecture on a vector of eleven features to predict Heart failure with preserved ejection fraction (HFpEF). For the detection of heart murmurs, an ELM was found to have performance similar to a support vector machine [54], and a Deep ELM network, or ELM with multiple hidden layers, was also successfully utilized for murmur detection [55]. Last, Ghosh et al. [56] propose a method that uses the ELM architecture, as an autoencoder, to reconstruct an image representation of heart sounds from a set of derived features, and the reconstructed image was then passed to a kernel sparse regression algorithm to classify any of five different diseases.

## VI. Conclusion

For the automated detection of pulmonary hypertension from heart sounds, we proposed PH-ELM, a novel PH detection algorithm based on the extreme learning machine, that generalizes reliably across the analyzed datasets and is computationally efficient. We also developed a rigorous evaluation methodology based on both in-distribution and out-of-distribution evaluations. Our results on three datasets show that two of four ML algorithms generalize well across PCG and SCG modalities, as well as from pigs to humans. To the best of our knowledge, this work is novel for its use of multiple datasets in the evaluation of Pulmonary Hypertension detection from heart sounds.

## References


[1] F. Valentin, *Hurst's the Heart, (2 Volume Set)*. New York, NY, USA: McGraw-Hill, 2011.

[2] M. Humbert et al., "2022 ESC/ERS guidelines for the diagnosis and treatment of pulmonary hypertension," *Eur. Respir. J.*, vol. 61, 2023, Art. no. 2200879.

[3] M. M. Hoeper et al., "Pulmonary hypertension," *Deutsches Ärzteblatt Int.*, vol. 114, no. 5, 2017, Art. no. 73.

[4] M. M. Hoeper et al., "A global view of pulmonary hypertension," *Lancet Respir. Med.*, vol. 4, no. 4, pp. 306–322, 2016.

[5] A. Frost et al., "Diagnosis of pulmonary hypertension," *Eur. Respir. J.*, vol. 53, no. 1, 2019, Art. no. 1801904.

[6] M. Ginoux et al., "Impact of comorbidities and delay in diagnosis in elderly patients with pulmonary hypertension," *ERJ Open Res.*, vol. 4, no. 4, 2018, Art. no. 00100-2018.

[7] E. M. Lau, M. Humbert, and D. S. Celermajer, "Early detection of pulmonary arterial hypertension," *Nat. Rev. Cardiol.*, vol. 12, no. 3, pp. 143–155, 2015.

[8] S. Janda et al., "Diagnostic accuracy of echocardiography for pulmonary hypertension: A systematic review and meta-analysis," *Heart*, vol. 97, pp. 612–22, 2011.

[9] J. Xu, L. Durand, and P. Pibarot, "A new, simple, and accurate method for non-invasive estimation of pulmonary arterial pressure," *Heart*, vol. 88, no. 1, pp. 76–80, 2002.

[10] N. Yamakawa et al., "Cardiac acoustic biomarkers as surrogate markers to diagnose the phenotypes of pulmonary hypertension: An exploratory study," *Heart Vessels*, vol. 37, pp. 593–600, 2022.

[11] J. Huang et al., "Noninvasive evaluation of pulmonary hypertension using the second heart sound parameters collected by a mobile cardiac acoustic monitoring system," *Front. Cardiovasc. Med.*, vol. 10, 2023, Art. no. 1292647.

[12] A. Dennis et al., "Noninvasive diagnosis of pulmonary hypertension using heart sound analysis," *Comput. Biol. Med.*, vol. 40, no. 9, pp. 758–764, 2010.

[13] T. Kaddoura et al., "Acoustic diagnosis of pulmonary hypertension: Automated speech-recognition-inspired classification algorithm outperforms physicians," *Sci. Rep.*, vol. 6, no. 1, 2016, Art. no. 33182.

[14] M. Wang et al., "Transfer learning models for detecting six categories of phonocardiogram recordings," *J. Cardiovasc. Develop. Dis.*, vol. 9, no. 3, 2022, Art. no. 86.

[15] A. Gaudio et al., "Explainable deep learning for non-invasive detection of pulmonary artery hypertension from heart sounds," in *Proc. 2022 Comput. Cardiol.*, 2022, pp. 1–4.

[16] B. Ge et al., "Detection of pulmonary arterial hypertension associated with congenital heart disease based on time–frequency domain and deep learning features," *Biomed. Signal Process. Control*, vol. 81, 2023, Art. no. 104451.

[17] A. N. Patnaik, "First heart sound," *Indian J. Cardiovasc. Dis. Women-WINCARS*, vol. 4, no. 02, pp. 107–109, 2019.

[18] J. M. Felner, "The Second Heart Sound. In: Walker HK, Hall WD, Hurst JW, editors," *Clin. Methods: Hist., Phys., Lab. Examinations*, 3rd edition. Boston: Butterworths, Chapter 23, 1990, Art. no. 122. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK341/

[19] F. Renna et al., "Separation of the aortic and pulmonary components of the second heart sound via alternating optimization," *IEEE Access*, vol. 12, pp. 34632–34643, 2024.

[20] N. J. Mehta and I. A. Khan, "Third heart sound: Genesis and clinical importance," *Int. J. Cardiol.*, vol. 97, no. 2, pp. 183–186, 2004.

[21] A. Gaudio et al., "Explainable deep learning method for non-invasive detection of pulmonary hypertension from heart sounds," Worldwide Patent WO 2024047610A1, Mar. 2024.

[22] A. Gaudio et al., "ExplainFix: Explainable spatially fixed deep networks," *WIREs Data Mining Knowl. Discov.*, vol. 13, no. 2, 2023, Art. no. e1483, [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1483

[23] A. Gaudio and M. Elhilali, "AP*: A modified average precision score and precision recall space for class Imbalanced datasets," TechRxiv, Oct. 2, 2024, doi: 10.36227/techrxiv.172788977.76284252/v1.

[24] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *Proc. 2013 Humaine Assoc. Conf. Affect. Comput. Intell. Interaction*, 2013, pp. 245–251.

[25] M. McDermott et al., "A closer look at auroc and auprc under class imbalance," *Adv. in Neural Inf. Process. Syst.*, A. Globerson et al., vol. 37, 2024, pp. 44102–44163. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/4df3510ad02a86d69dc32388d91606f8-Paper-Conference.pdf

[26] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[27] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, New York, NY, USA, 2006, pp. 233–240, [Online]. Available: https://doi.org/10.1145/1143844.1143874

[28] A. Gaudio et al., "Cross-domain detection of pulmonary hypertension in human and porcine heart sounds," in *Proc. Comput. Cardiol.*, 2023, vol. 50, pp. 1–4.
[29] M. Chan et al., "SCG-RHC: Wearable seismocardiogram signal and right heart catheter database," *PhysioNet*, 2023. [Online]. Available: https://doi.org/10.13026/133d-pk11
[30] F. Renna, J. Oliveira, and M. T. Coimbra, "Deep convolutional neural networks for heart sound segmentation," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 6, pp. 2435–2445, Nov. 2019.
[31] N. Giordano et al., "A wearable multi-sensor array enables the recording of heart sounds in homecare," *Sensors*, vol. 23, no. 13, 2023, Art. no. 6241.
[32] G. Huang et al., "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[33] X. Liu, C. Gao, and P. Li, "A comparative analysis of support vector machines and extreme learning machines," *Neural Netw.*, vol. 33, pp. 58–66, 2012.
[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations {ICLR}*, New Orleans, LA, USA, May 2019. [Online]. Availabel: https://openreview.net/forum?id=Bkg6RiCqY7
[35] M. Andreux et al., "Kymatio: Scattering transforms in python," *J. Mach. Learn. Res.*, vol. 21, no. 60, pp. 1–6, 2020.
[36] Y.-Y. Yang et al., "Torchaudio: Building blocks for audio and speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 6982–6986.
[37] K. J. H. Kwak Sang Gyu, "Central limit theorem: The cornerstone of modern statistics," *Korean J. Anesthesiol.*, vol. 70, no. 2, pp. 144–156, 2017, [Online]. Available: http://www.e-sciencecentral.org/articles/?scid=1156667
[38] D. Teney et al., "ID and OOD performance are sometimes inversely correlated on real-world datasets," in *Proc. Adv. Neural Inf. Process. Syst.*, New Orleans, Louisiana, USA, 2023, vol. 36, pp. 71703–71722. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/e304d374c85e385eb217ed4a025b6b63-Paper-Conference.pdf
[39] A. D'Amour et al., "Underspecification presents challenges for credibility in modern machine learning," *J. Mach. Learn. Res.*, vol. 23, no. 226, pp. 1–61, 2022, [Online]. Available: http://jmlr.org/papers/v23/20-1335.html
[40] W. Ye et al., "Spurious correlations in machine learning: A survey," 2024, *arXiv:2402.12715*.
[41] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231206000385
[42] G. Huang et al., "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, 2015.
[43] J. Wang et al., "A review on extreme learning machine," *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 41611–41660, 2022.
[44] Y. Huérfano-Maldonado et al., "A comprehensive review of extreme learning machine on medical imaging," *Neurocomputing*, vol. 556, 2023, Art. no. 126618.
[45] M. Nahiduzzaman et al., "Diabetic retinopathy identification using parallel convolutional neural network based feature extractor and ELM classifier," *Expert Syst. Appl.*, vol. 217, 2023, Art. no. 119557.
[46] S. Zhou and B. Tan, "Electrocardiogram soft computing using hybrid deep learning CNN-ELM," *Appl. Soft Comput.*, vol. 86, 2020, Art. no. 105778.
[47] Z. Bai, L. L. C. Kasun, and G.-B. Huang, "Generic object recognition with local receptive fields based extreme learning machine," *Procedia Comput. Sci.*, vol. 53, pp. 391–399, 2015.
[48] J. Zhang et al., "An automatic recognition method of microseismic signals based on EEMD-SVD and ELM," *Comput. Geosciences*, vol. 133, 2019, Art. no. 104318.
[49] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, "PCA-ELM: A robust and pruned extreme learning machine approach based on principal component analysis," *Neural Process. Lett.*, vol. 37, pp. 377–392, 2013.
[50] J. Bruna, "Scattering representations for recognition," Ph.D. dissertation, Ecole Polytechnique X., Palaiseau, France, 2013.
[51] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
[52] A. Gaudio et al., "DeepFixCX: Explainable Privacy-Preserving Image Compression for Medical Image Analysis," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 13, no. 4, 2023, Art. no. e1495.
[53] Y. Liu, X. Guo, and Y. Zheng, "An automatic approach using ELM classifier for HFpEF identification based on heart sound characteristics," *J. Med. Syst.*, vol. 43, no. 9, 2019, Art. no. 285.
[54] X. Yang et al., "A multi-modal classifier for heart sound recordings," in *Proc. 2016 Comput. Cardiol. Conf.*, 2016, pp. 1165–1168.
[55] P. R. Malaysia and B. Pahat, "An innovative machine learning framework for phonocardiography (PCG) using MFCC and deep extreme learning machine (DELM)," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 22, 2024.
[56] S. K. Ghosh et al., "Deep layer kernel sparse representation network for the detection of heart valve ailments from the time-frequency representation of PCG recordings," *BioMed Res. Int.*, vol. 2020, no. 1, 2020, Art. no. 8843963.