Supplemental Online Content

Laumer F, Di Vece D, Cammann VL, et al. Assessment of artificial intelligence in echocardiography diagnostics in differentiating takotsubo syndrome from myocardial infarction. *JAMA Cardiol.* Published online March 23, 2022. doi:10.1001/jamacardio.2022.0183

This supplemental material has been provided by the authors to give readers additional information about their work.

eAppendix. Model architecture and implementation details

Convolutional autoencoder for feature extraction

The layer-by-layer description of the neural network architecture of the encoder and decoder is presented in the table below. The inputs are semantically segmented echocardiographic video frames. Each pixel is represented as a one-hot vector of dimensionality 4 for the 2-chamber view, and dimensionality of 6 for the 4-chamber view, respectively.

Encoder	
Layer type	**Output shape (2-ch. / 4-ch.)**
Input (segmented frame, one-hot pixel vector)	112x112x4 / 112x112x6
Conv. 4x4 kernel size, 16 filters, stride 2, ReLU	55x55x16
Conv. 4x4 kernel size, 16 filters, stride 2, ReLU	26x26x16
Conv. 4x4 kernel size, 32 filters, stride 2, ReLU	12x12x32
Conv. 4x4 kernel size, 32 filters, stride 2, ReLU	5x5x32
Flatten	800
Fully-connected, 16 latent nodes, linear	16 (latent nodes)

Decoder	
Layer type	**Output shape (2-ch. / 4-ch.)**
Input (compressed frame representation)	16 (latent nodes)

Fully-connected. 800 units, ReLU	800
Reshaping	5x5x32
Transp. conv. 4x4 kernel size, 32 filters, stride 2, ReLU	12x12x32
Transp. conv. 4x4 kernel size, 16 filters, stride 2, ReLU	26x26x16
Transp. conv. 4x4 kernel size, 16 filters, stride 2, ReLU	55x55x16
Transp. conv. 4x4 kernel size, 4/6 filters, stride 2, ReLU	112x112x4 / 112x112x6

We trained the neural network on the artificially augmented training dataset by backpropagation using the Adam optimizer with a learning rate of 0.0005. We run 100 steps per epoch with a mini-batch size consisting of 64 frames. 20% of the data were used for early stopping and we stopped training if the loss did not decrease during the last 20 epochs. We used a weighted linear combination of the mean squared reconstruction loss and cross-entropy-loss as our loss function. The trained autoencoder model is able to extract disease informative features and furthermore removes artefacts present in the input data and outputs smooth, and more likely semantic segmentations.

Temporal neural network for sequence classification

We used a fully convolutional neural network architecture based on 1D convolutional layers for the sequence encoders for the 2-chamber and the 4-chamber view as described in the tables below. The output of these encoder network is concatenated and we used a fully connected layer with a Softmax activation function to output the disease

probabilities. The layer-by-layer description of the neural network is reported in the table below. The input sequences were first resampled to a fixed number of 70 time steps. The same artificially augmented dataset as for the autoencoder training was used for training the temporal neural network. Additionally, 10 random subsequences of length 70 were extracted from each sequence that was larger than 70 time steps.

Sequence encoder (2-ch)	
Layer type	**Output shape**
Input (extracted sequences, time steps x latent nodes)	70x16
1D Conv. 3 kernel size, 32 filters, stride 1, ReLU	68x32
Average pooling, pool size 3	22x32
1D Conv. 3 kernel size, 32 filters, stride 1, groups 8, ReLU	20x32
Average pooling, pool size 3	6x32
1D Conv. 3 kernel size, 32 filters, stride 1, ReLU	4x32
Average pooling, pool size 3	1x32
Flatten	32

Sequence encoder (4-ch)	
Layer type	**Output shape**

Input (extracted sequences, time steps x latent nodes)	70x16
1D Conv. 3 kernel size, 8 filters, stride 1, ReLU	68x8
Average pooling, pool size 3	22x8
1D Conv. 3 kernel size, 8 filters, stride 1, groups 8, ReLU	20x8
Average pooling, pool size 3	6x8
1D Conv. 3 kernel size, 16 filters, stride 1, ReLU	4x16
Average pooling, pool size 3	1x16
Flatten	16

In the first two 1D convolutional layers we used kernel, bias and activity regularizer. We trained the neural network by backpropagation using the Adam optimizer with a learning rate of 0.0005. We used a mini-batch size consisting of 64 samples and 100 steps per epoch. 20% of the data was used for early stopping, and we stopped training if the loss did not decrease during the last 20 epochs. We used categorical cross-entropy as loss function.

For the experiments where only one chamber was used as input, the sequence encoder architecture for the respective chamber was used and only the classification layer was adapted to take one chamber feature input instead of two.

Statistical analyses were performed using Python 3.7 and IBM SPSS Statistics, version 25.0. The machine learning models were developed and tested in Tensorflow 2.2.

Pitfalls and important lessons learned

Developing a machine learning pipeline on real world data is a challenging task. Especially, in clinical practice where standardized protocols for data acquisition have not yet fully established. For example, echocardiography images are influenced by many factors of variations, e.g. resolution, scanner model and software settings. Those factors influence the video's grey value distribution giving rise to potential confounding factors. Machine learning models are very good in learning so-called *shortcuts*. Shortcuts are predictions made on features that are not causally related to the disease. For example, if all the TTS samples are brighter than the AMI samples a machine learning algorithm would pick up this shortcut and start basing its decision on this feature, which is unrelated to the underlying disease. In our first iteration of the algorithm, this was the case. We had developed a similar classification algorithm as the one presented in this manuscript. We used one dataset of 220 Patients (110 AMI and 110 TTS) for training and evaluation. We trained the algorithm on raw (not segmented) frames and evaluated its performance via nested cross-validation. The performance was quite good and achieving a mean ROC-AUC of 0.801 with an overall mean accuracy of 74.5% calculated over 5 different validation folds.

We acquired a second independent dataset consisting of 228 Patients (114 AMI and 114 TTS). We retrained the algorithm on the full original dataset and tested it on the new dataset. The performance dropped drastically, achieving a ROC-AUC between 0.38 and 0.52, and accuracies between 44% and 54% when evaluated over different training runs with different seeds. Training on the new dataset and evaluating on the original one,

resulted in similarly bad scores with ROC-AUC ranging from 0.37 to 0.50 and accuracies between 41% and 53%.

The algorithm had learned a shortcut to classify the patients. Confounders introduced by the different ultrasound scanner models were not removed by cropping away scanner meta data nor by data standardization. Below, in eTable 1, the scanner models for each cohort are listed.

eTable. Scanner Models for Each Cohort

Is it possible to observe that the scanner models within one cohort correlate with the disease labels. However, the disease per scanner distribution is quite different in the other cohort

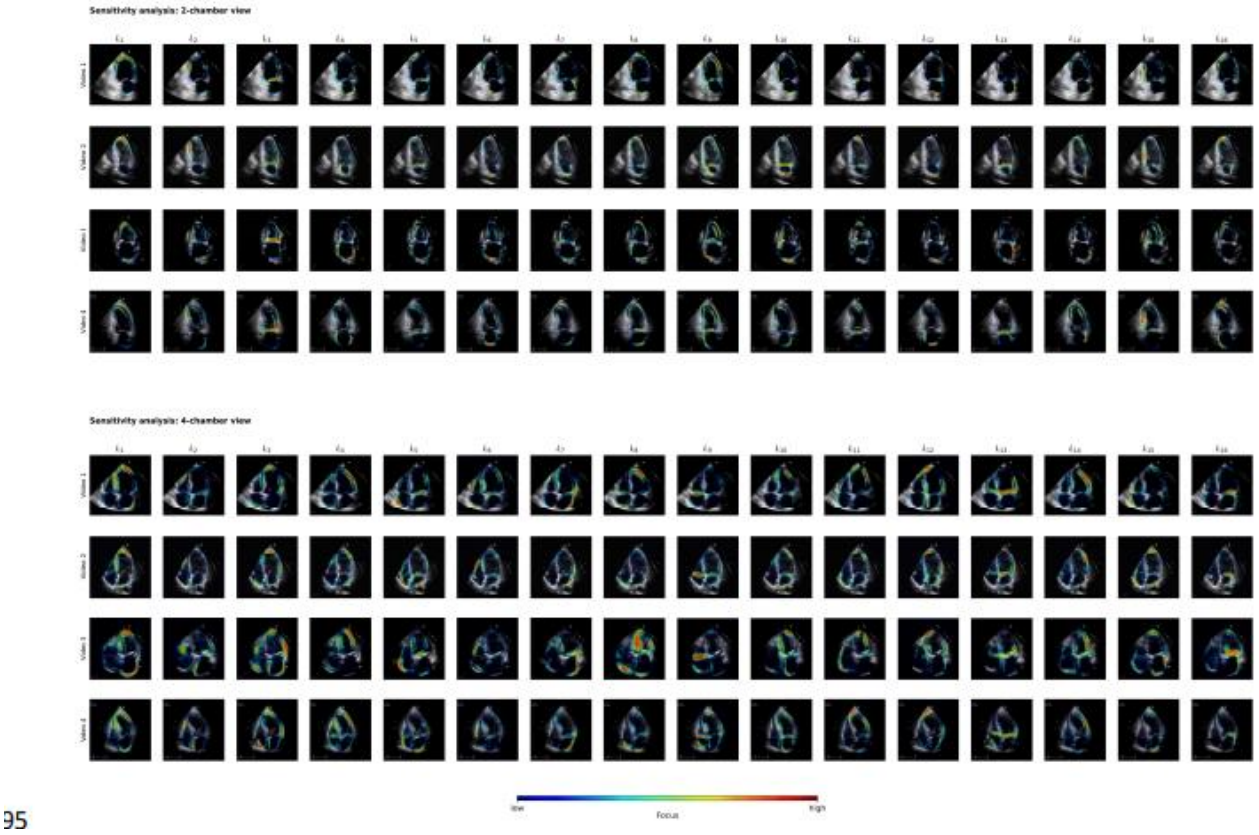
Scanner Model	Training Cohort		Test Cohort	
	AMI	TTS	AMI	TTS
iE33	7	70	75	35
Vivid E9	12	18	12	10
Vivid E95	33	10	0	38
Vivid 7	0	0	7	8
Vivid q	0	0	1	0
EPIQ 7C	57	16	5	12
Sequoia	1	0	2	2
CX50	3	0	1	0
TUS AI900	1	0	0	0
unknown	0	0	7	5
SUM	114	114	110	110

The original model, besides learning disease specific patterns, also picked up scanner model specific features to classify the echocardiography videos into AMI and TTS. The different distribution of scanner models in the training and test cohorts explains the massive drop in performance. As a solution, we semantically segmented the videos and trained the machine learning model on the segmentation maps. The segmentation process

removed any confounders present in the grey value distribution of the echocardiography video. While this segmentation process removes scanner specific characteristics it also removes information about the tissue composition that might be relevant for diagnosis. Furthermore, extensive data augmentation both during training and inference time, as described in the main document, were necessary to achieve satisfactory results.

Echocardiography-based diagnosis of TTS vs AMI by trained cardiologists were only available for the 220 patients of the original dataset. As we wanted to compare the performance in image-based disease diagnosis of the machine learning algorithm to medical experts, we decided to develop and train the new confounder free model on the newer dataset of 228 patients and to evaluate it on the original dataset to enable direct comparison of performance between the two entities.

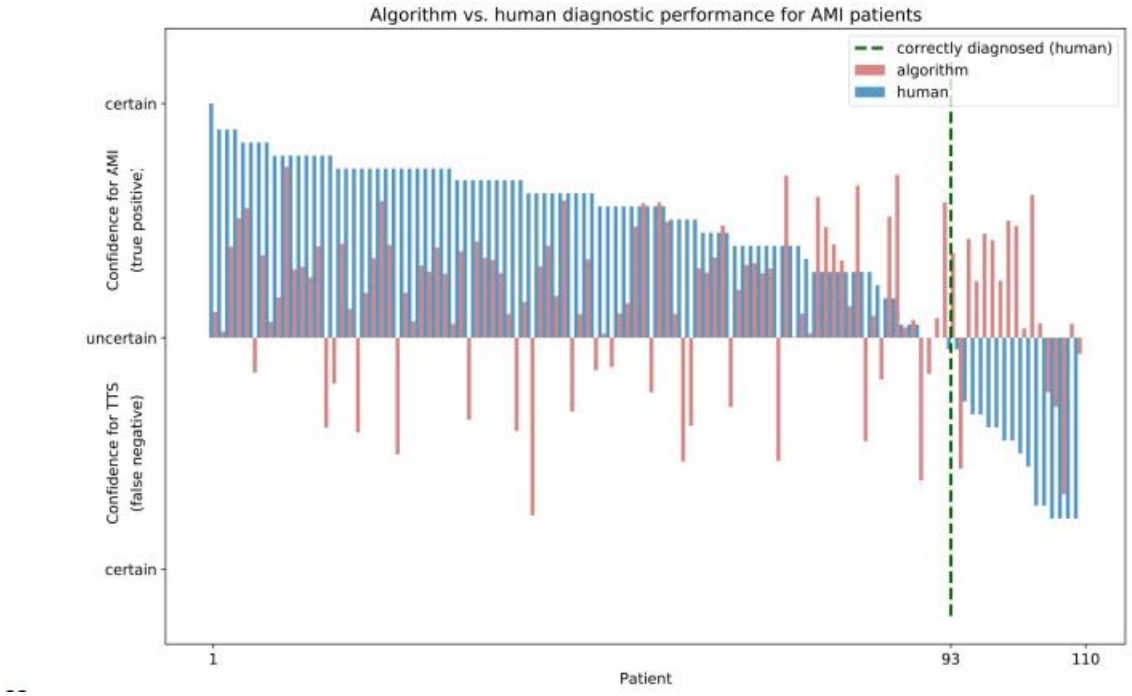
Supplementary figures



95

eFigure 1. Result of the sensitivity analysis of the different latent feature nodes

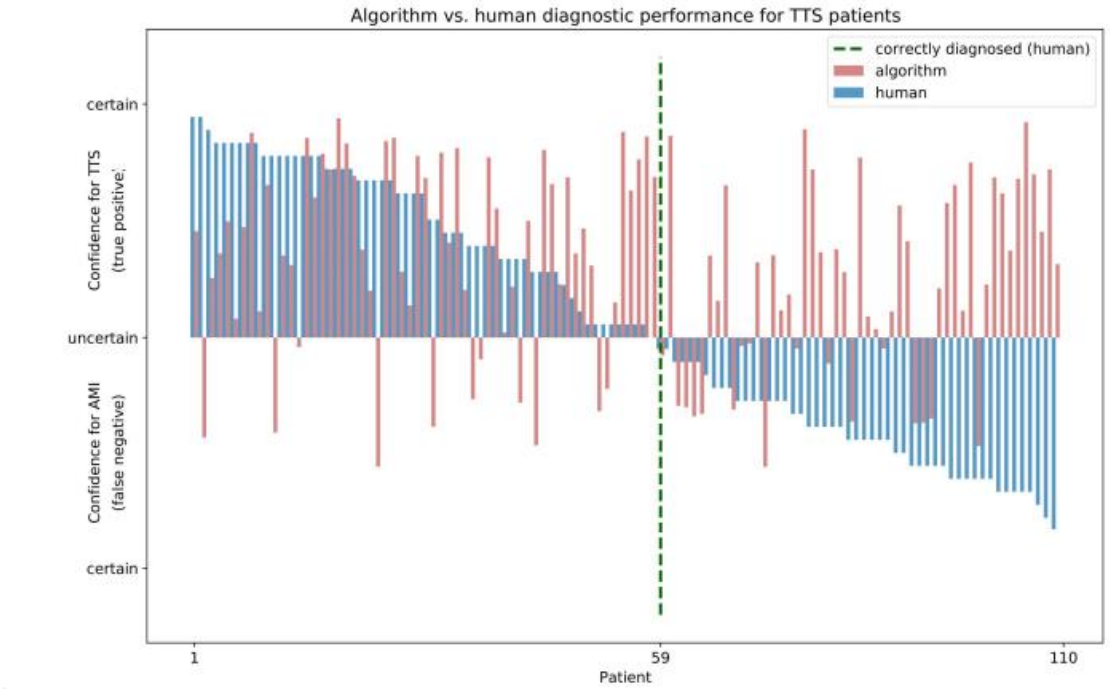
The result of the sensitivity analysis of the 16 different latent feature values ($L_1 - L_{16}$) is visualized in the raw echocardiogram. The top four rows present a random selection of 2-chamber view videos and the bottom rows show a random selection of 4-chamber view videos. One can see how the different latent feature values ($L_1 - L_{16}$) focus on different regions within the echocardiogram. These regions are consistent across the different videos.



eFigure 2. Per patient diagnostic performance of AMI (humans vs algorithm)

The cardiologists specified on a 0 to 9 integer scale their confidence that a particular patient has TTS. While 9 corresponds to 100% confidence for TTS, 0 corresponds to 100% confidence for AMI. This means that predictions 0-4 correspond to diagnosing AMI and predictions 5-9 correspond to diagnosing TTS. This plot shows the diagnostic performance of the cardiologists for all AMI patients. The predictions from the four cardiologists where averaged per patient and sorted according to their stated confidence that a particular patient suffers from AMI. For the figure, the cardiologists' predictions were rescaled between -1 and 1, with -1 corresponding to 9 (meaning 100% confidence for TTS) and 1 corresponding to 0 (meaning 100% confidence for AMI). 93 out of 100

patients with AMI were correctly diagnosed by the medical experts. In the cases where cardiologists decided against AMI, considering the machine prediction would reduce the number of false negatives to only 5 out of 110 (visible on the right hand side of the plot after the green dotted line).



eFigure 3. Per patient diagnostic performance of TTS (humans vs algorithm)
This plot shows the diagnostic performance of the cardiologists for all TTS patients. The predictions from the four cardiologists were averaged per patient and sorted according to their stated confidence that a particular patient has TTS. For the figure, the cardiologists' predictions were rescaled between -1 and 1, with -1 corresponding to 0 (meaning 100% confidence for AMI) and 1 corresponding to 9 (meaning 100% confidence for TTS). The cardiologists identified only 59 out of 110 TTS patients correctly.