A Study on Several Critical Problems on Arrhythmia Detection using Varying-Dimensional Electrocardiography

Jingsu Kang[1], Hao Wen[2]

[1]Tianjin Medical University, Tianjin, China
[2]LMIB, School of Mathematical Sciences, Beihang University, Beijing, China

Abstract

As a major tool for the diagnosis of heart diseases, the electrocardiography (ECG) has long been an active research field among medical researchers. Among various types of ECGs, the standard 12-lead ECG is more extensively studied, due to its clinical significance. In order to mitigate the workload of the cardiologists, improving the efficiency, accuracy and availability of this diagnostic tool, there has been a continuous effort in the ECG research community to design algorithms and systems for automatic analysis of multi-dimensional ECGs. Although many excellent algorithms have been developed and a number of successful systems have already been applied in clinical, there are still fundamental problems that have not yet been thoroughly studied and fully answered.

The PhysioNet/Computing in Cardiology Challenge 2021 (CinC2021) raised several such critical problems: can subsets of the standard 12 leads provide models with adequate information to give comparable performances for classifying ECG abnormalities? can models be designed to be effective enough to classify a broad range of ECG abnormalities? In this work, we present our effort on tackling these problems. Through extensive searching, we discover several neural network architectures that provide such effectiveness and moreover they provide comparable performances on reduced-lead ECGs, even in the extreme case of 2-leads. To further augment model performances on specific ECG abnormalities and to improve interpretability, we manually design auxiliary detectors based on clinical diagnostic rules. To conclude, our work gives positive answer to the critical questions CinC2021 raises and lays solid foundation for further research in the future on these topics.

Keywords: multi-dimensional ECG, reduced leads, deep learning, neural architecture search, clinical rule based detector

1. Introduction

Heart disease is the leading cause of death worldwide [1]. Electrocardiogram (ECG), as a physiological signal that reflects the electrical activity of the myocardium, is widely adopted for screening heart diseases in clinic. Despite the rapid growth of the application of ambulatory ECG and other methods, the standard 12-lead ECG is still the most widely accepted in the clinical practice. However, the 12-lead ECG equipments for real-time monitoring of the cardiac electrical activities, especially the 24-hour Holter monitoring, usually collect an enormous amount of data. Therefore, how to properly process these ECG signals, so as to be rapid, accurate and complete for the diagnosis of a range of cardiac abnormalities as broad as possible, is still an urgent problem that needs to be solved.

For this reason, many algorithms have been proposed to accurately and automatically assist in the diagnosis of cardiac electrical abnormalities. Among them, deep neural networks (DNNs) have achieved great success [2–4] in recent years and have been dominating in this research field. These models are claimed to be able to achieve very high precision, even comparable to cardiologists [2]. However, these work have their drawbacks. For example, their models have only been validated on just a few typical ECG arrhythmias (for example atrial fibrillation, ventricular tachycardia, etc. [2]). Data redundancy is another aspect that these work did not consider [3, 4]. Hence the problem raised in the begging of the paper still needs to be studied in more depth.

In this paper, our effort of tackling this problem, which uses DNNs in combination with clinical rules providing better interpretability, will be described. Following the mission of The PhysioNet/Computing in Cardiology Challenge 2021 (CinC2021) [5–7], we apply our hybrid method on the problem of classifying a very broad range of cardiac abnormalities, instead of just several typical ones, using reduced-lead ECGs which largely reduce the data amount while at the cost of only holding partial information of the standard 12-lead ECGs. The 5 lead-sets reduced from the standard 12-lead are listed as follows.

- Twelve leads: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6
- Six leads: I, II, III, aVR, aVL, aVF
- Four leads: I, II, III, V2
- Three leads: I, II, V2
- Two leads: I, II

We find that our method can achieve performances comparable to the standard 12-lead ECGs on the reduced-lead ECGs, even in the extreme 2-lead case. This might suggest that the standard 12-lead ECGs might be highly redundant for the diagnosis of a very large proportion of the cardiac abnormalities. Another point is that our DNN models have acceptable performance on the detection of this wide range of cardiac abnormalities, hence can serve as eligible assists for the cardiologists.

Our findings will be introduced in more details in the rest of the paper, which is organized as follows: Section 2 explains the selection of data processing methods, including methods of data preprocessing and data augmentation. Section 3 introduces the deep learning models with novel network architectures. Section 4 that follows discusses the training setups for the neural network models, including the selection of loss functions, optimizers and learning rate schedulers, etc. In order to assist the deep learning models, we design several detectors based on clinical rules for easily distinguishable abnormal ECG signals in section 5. The performances of our solution to the CinC2021 challenge problems are summarized in section 6. Section 7 contains our thorough ablation studies for the selection of model structures, loss functions and data augmentation methods, etc. Section 8 concludes this work.

2. Data Processing

Most ECG abnormalities are characterized by their subtle changes both morphological and spectral, while only a minority can be described easily by clinical diagnostic criteria, for example bradycardia, low QRS voltage, etc. DNNs which are particularly suitable for capturing such subtle structures in a neat and uniform way become the de facto default method for various ECG tasks, as stated in the previous section. It is commonly believed that DNNs are robust to “natural”[1] noises, hence require little preprocessing as compared to traditional signal processing methods. Instead of traditional preprocessing techniques, the deep learning research community tend to use data augmentation techniques to broaden the distribution of the training data, which in theory help improve performances of DNNs on unseen data. However, to the authors’ knowledge, very few numerical results can be found in ECG deep learning research work on this topic. Therefore, we include both traditional preprocessing techniques as well as data augmentation techniques which we think would be helpful and suitable for developing our deep neural networks. A simple numerical comparison will be given in Section 7. Since we also designed clinical rule based detectors, hence the inclusion of traditional preprocessing is reasonable.

2.1. Data Selection

The main data resource of this study for conducting the experiments is the public training data of CinC2021 [7], excluding the StPetersburg subset (the INCART dataset [8]) since the granularity of its labels is too coarse, being not suitable for training neural networks. Beside the INCART dataset, the public part of this database contains 4 other subsets, namely

- CPSC Database and CPSC-Extra database [9]
- PTB [10] and PTB-XL [11] database
- Georgia 12-lead ECG Challenge (G12EC) database [6]
- Chapman-Shaoxing [12] and Ningbo [13] database

There are totally 85056 records of 132 classes (abnormalities) in the CinC2021 database. We use only the 26 scored classes (equivalent classes counted as 1) in the challenge which cover 92.46% of the whole database.

2.2. Preprocessing Procedures

To train the neural networks, the ECGs are resampled to 500 Hz if necessary. ECGs are further cropped or padded by zeros to a length of 10 seconds (5000 sample points) in order to utilise mini-batch (parallel) training.

It is a common practice in traditional ECG signal processing to filter the ECGs with a band-pass filter to remove noises as well as coarse baseline drifts. Sometimes an additional median filter is used to further remove the baseline. Another advantage of band-pass filtering is that such operation can largely bridge the potential gap between training and inference data distributions for DNNs. In almost all our experiments, we take band-pass filtering, using a finite impulse response (FIR) filter with passband 0.5 Hz - 60 Hz, as our initial step of the data processing pipeline. To validate its efficacy for DNNs, experiments are performed as a part of our ablation study in Section 7.1.

Another preprocessing procedure we take is the z-score normalization defined as follows

$$\frac{\mathbf{x} - \mathrm{mean}(\mathbf{x})}{\mathrm{std}(\mathbf{x}) + \mathbf{eps}} \cdot s + m$$

where $\mathbf{x}$ is the ECG signal, s, m, eps are fixed values. In our experiments, we take $s = 1, m = 0, eps = 10^{-7}$. The small value eps is added to avoid division by zero for those ECG records with constant value. After this transformation, the ECG will have mean m and standard deviation s (except for constant value ECGs). Numerical results

[1] in contrast to man-made adversarial attack

verifying the validity of z-score normalization are also presented in Section 7.1. As z-score normalization has similar effect to a median filter, the latter is often omitted when preparing data for deep learning models.

2.3. Data Augmentation

To alleviate overfitting, data augmentation techniques including `mixup` [14], random masking [3] are adopted stochastically (with certain probability) on the training data for the neural networks. `mixup` performs convex linear combination of the training data as follows

$$\begin{aligned} \mathbf{x} &= \alpha \mathbf{x_1} + (\mathbf{1} - \alpha)\mathbf{x_2}, \\ \mathbf{y} &= \alpha \mathbf{y_1} + (\mathbf{1} - \alpha)\mathbf{y_2}, \end{aligned}$$

where $0 < \alpha < 1$, $\mathbf{x_1}, \mathbf{x_2}$ are preprocessed ECGs, and $\mathbf{y_1}, \mathbf{y_2}$ are corresponding labels, and the input for the neural network is their convex linear combination $\mathbf{x}, \mathbf{y}$. No more augmentations, like random flip, are done, since they might completely change the interpretation of the standard-lead ECGs[2], as will be seen in Section 5. To suppress overconfidence which could help improving generalization capabilities of models, the technique of label-smoothing regularization (LSR) [15] is used. Let $\mathbf{y}$ be an one-hot label vector (the "hard" label), then LSR generates "soft" label vector via Equation (1)

$$\mathbf{y}' = (\mathbf{1} - \varepsilon)\mathbf{y} + \frac{\mathbf{1}}{\mathbf{K}}\varepsilon \mathbf{e}, \tag{1}$$

where K is the number of classes, $\mathbf{e}$ is the K-dimensional vector with all entries are 1, and $\varepsilon \in [0, 1]$ is a weight factor. In our approaches, we take $\varepsilon = 0.1$.

Similar to preprocessing procedures, we performed experiments validating the impact on model performances of the augmentation techniques in Section 7.1.

3. Neural Network Architectures

3.1. Convolutional Recurrent Neural Networks (CRNNs)

Inspired by previous work [3, 16], we base our pipeline for solving ECG classification problems on a CRNN framework. A CRNN consists of a CNN backbone extracting features from input ECGs. Following the CNN backbone is an optional RNN module (e.g. LSTM [17]) and an optional attention module (e.g. squeeze-and-excitation (SE) [18], global context (GC) [19]) to refine the feature maps, modeling long-range dependencies and making use of the sequence properties of ECGs. Thus obtained feature maps are fed into a multi-layer (typically 1 layer) perceptron (MLP) head to produce the vector of probabilities for each class.

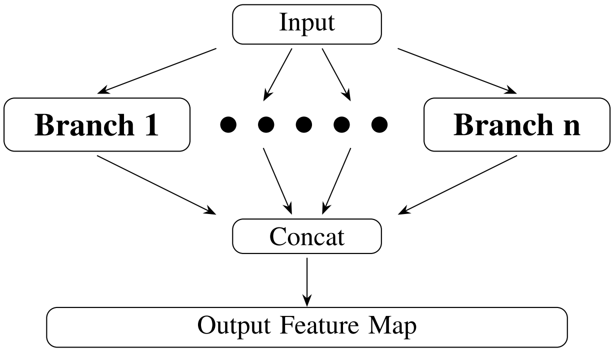


Figure 1: Sketch of the architecture of the multi-branch CNN family.

The CNN backbone is the core of this CRNN framework and has been extensively studied by the ECG research community. Variants [2–4] from computer vision models, as well as networks distinctive for ECGs [16], have enjoyed great success in many ECG tasks. Design and selection of CNN backbones will be the central issue for this study.

3.2. Multi-branch CNNs

The first set of CNN backbones we use for solving the CinC2021 challenge problem are derived from the multi-branch CNNs proposed in [16], whose general architecture in depicted in Figure 1. This type of network has several branches, with each branch containing sequentially stacked convolutional layers, and with normalization layers, activation layers and dropout layers inserted after specific convolutional layers. The essence of this network family is that each branch uses dilated convolutions with different dilation ratio, so that the receptive field of each branch matches specific waveforms of the ECG. The idea behind is similar to the ASPP (Atrous Spatial Pyramid Pooling) head originally proposed in DeepLabV2 [20], directly modelling the spectral characteristics. We used a 3-branch CNN backbone in our official phase submission entry for the CinC2021 challenge, whose key hyperparameters are listed in Table 1.

3.3. "Lead-wise" CNNs

Another novel structure of the CNNs we design during developing our solution to the CinC2021 challenge problem is the "lead-wise" CNNs, which are implemented via grouped convolutions so that the number of groups are divisible by the number of leads of the input ECGs. Another

[2] which might not be the case for wearable dynamic ECGs

	Kernel Sizes	Dilations
Branch 1	$(11, 7, 7, 5, 5, 5)$	$(1, 1, 1, 1, 1, 1)$
Branch 2		$(2, 2, 4, 8, 8, 8)$
Branch 3		$(4, 4, 8, 16, 32, 64)$

Table 1: Kernel sizes and dilations for the convolutions for a typical example of the CNN described in Figure 1. It has three branches, each with 6 convolutional layers. We used this CNN backbone in our official phase submission entry for the CinC2021 challenge.

key point is that normalization layers should use group normalizations [21] instead of batch normalization. Using this mechanism, training one model on multi-lead ECGs is equivalent to training multiple models on single-lead ECGs. This enables parameter reuse for training models on reduced-lead ECGs from models trained on the standard 12-lead ECGs, in which case one freezes the parameters of the CNN backbone and fine-tunes the rest part of the network. The "lead-wise" CNNs have significantly less parameters compared to their "normal" CNN counterparts, while offering competitive performances. This would be shown via experiments in Section 7.2.

3.4. Variants of ResNet

For the purpose of searching for more effective model architectures other than the networks we used for the CinC2021 challenge, we experiment with various ResNet variants (e.g. [2, 4, 22, 23]) as CNN backbone for our neural networks. The original ResNet was proposed in [24] in 2016, and quickly became the most widely used neural network architecture. The TResNet [23] variants currently are still almost the state-of-the-art (SOTA) model for multi-label classification on the MS-COCO dataset [25]. Until recently, variants of ResNet [2,4] have long been dominating the research community of ECG processing using deep learning.

A typical architecture of ResNet consists of an input stem, followed by 4 stages, as depicted in Figure 2. The stem consists of convolutions with large kernel sizes and with [22–24] or without [4] down-sampling layers. The 4 stages consist of different numbers of stacked building blocks of basic type (Figure 3a) or of bottleneck type (Figure 3b). Attention mechanisms, which are aforementioned in Section 3.1, are usually integrated into these building blocks to further improve the performance of the networks.

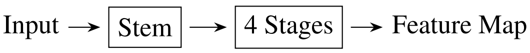
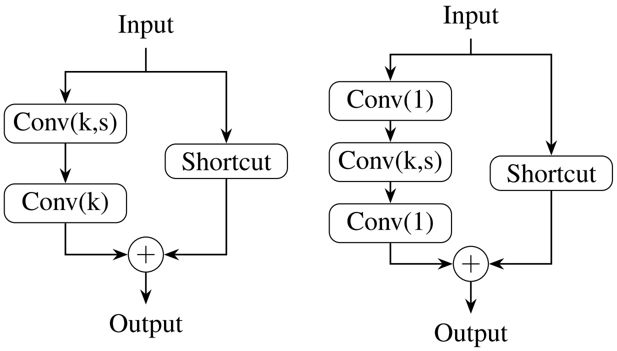


Figure 2: A typical architecture of ResNet. The Stem typically consists of convolutions with large kernel sizes to capture coarsely-level features and down-sampling layers to reduce computations. The 4 stages consists of stacks of units sketched in Figure 3 producing the feature maps for final classification.



(a) Basic Block

(b) Bottleneck Block

Figure 3: Sketch of 2 types of building blocks for ResNet. Conv(k,s) is convolution of kernel size k, and stride s; Conv(1) is convolution of kernel size 1 and stride 1. When $s = 1$, the shortcut is just the identity map; when $s > 1$, shortcut need to do down-sampling. When the main stream (the path passing from Input to Output through the convolutions) raises the number of channels, shortcut also needs to use convolutions of kernel size 1 to raise to the same number of channels. Other layers including activations, normalizations and dropouts are not plotted in this Figure. In (b), it is actually from the ResNet-B variant, rather than the original one.

The SE-variant (with SE attention integrated in the building blocks) of the ResNet model proposed in [4] will serve as the baseline for our comparative studied. This baseline will be denoted as ResNet-NC-SE ("NC" is short for "Nature Communications"). Its bottleneck variant, denoted as ResNet-NC-BS, along with another variant ResNet-NC-BG, obtained by replacing the SE attention module with GC attention module in ResNet-NC-BS, are also experimented.

Another series tested is the TResNet family [23]. TResNets adopted many new techniques for acceleration and/or improving prediction accuracy and robustness. These techniques include space-to-depth (S2D) [26] operation, anti-aliasing (AA) down-sampling [27], Since the smallest network (the TResNet-M) proposed therein has 21 building blocks, which is much too large for ECG processing, we design and experiment with 3 smaller variants, namely TResNet-N (nano), TResNet-P (pico), TResNet-F (femto). More details on architectures of the above networks are gathered in Table 2 and Table 3. Experiment results will be presented in Section 7.

Network	Stem		Stage1		Stage2		Stage3		Stage4	
	type	chan	repeat, type	chan	repeat, type	chan	repeat, type	chan	repeat, type	chan
`ResNet-NC-SE`	Conv	64	1 Basic-SE	128	1 Basic-SE	192	1 Basic-SE	256	1 Basic-SE	320
`ResNet-NC-BS`	Conv	64	1 Bottle-SE	512	1 Bottle-SE	768	1 Bottle-SE	1024	1 Bottle-SE	1280
`ResNet-NC-BG`	Conv	64	1 Bottle-GC	512	1 Bottle-GC	768	1 Bottle-GC	1024	1 Bottle-GC	1280
`TResNet-N`	S2D	56	2 Basic-SE	56	2 Basic-SE	112	2 Bottle-SE*	896	2 Bottle	1792
`TResNet-P`	S2D	56	1 Basic-SE	56	1 Basic-SE	112	1 Bottle-SE*	896	1 Bottle	1792
`TResNet-F`**	S2D	32	1 Basic-SE	32	1 Basic-SE	64	1 Bottle-SE*	512	1 Bottle	1024

Table 2: Architectures of Variants of ResNet. Types and number of repeats and output channels of the building blocks used are listed. For the stem, “Conv” means one convolutional layer with kernel size 17 and without down-sampling; “S2D” refers to the space-to-depth operation, possibly followed by a convolution with kernel size 1 to match the number of output channels of the stem.
* SE module follows the 2nd convolution of the block, the rest follows the last convolution of the block.
** Convolutions are separable convolutions.

Network	Kernel Size		Down-Sampling	
	Stem	4 Stages	Stem	4 Stages
`ResNet-NC-SE`	17	17	1	$(4,4,4,4)$
`ResNet-NC-BS`	17	17	1	$(4,4,4,4)$
`ResNet-NC-BG`	17	17	1	$(4,4,4,4)$
`TResNet-N`	1	15	4	$(1,2,2,2)$
`TResNet-P`	1	17	4	$(1,2,2,2)$
`TResNet-F`	1	17	4	$(1,2,2,2)$

Table 3: Details of the kernel sizes and down-sampling ratios of the stem and the building blocks of the 4 stages in the ResNet variants.

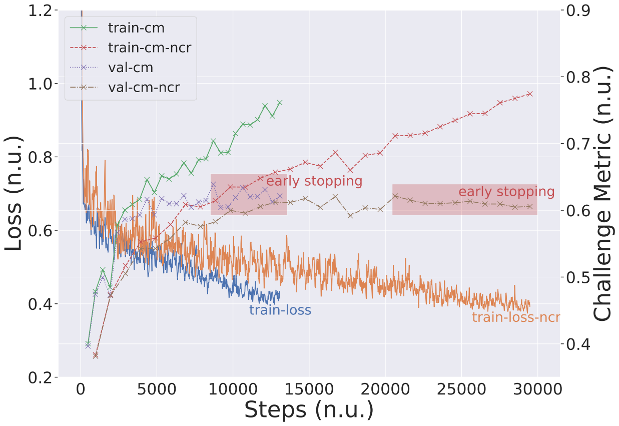


Figure 4: Two typical training processes using weighted BCE loss for neural networks with lead-wise branched CNN backbone. The suffix “ncr” refers to the challenge solution pipeline with **n**o **c**linical **r**ule based detector (ref. Section 5), in which case the whole 26 classes are included for training. “cm” is the abbreviation of the **c**hallenge **m**etric.

4. Training Setups

The whole neural network development and validation work are done on a GPU server with 8×RTX3080. In most cases, we set batch size to be 64 and set the maximum number of epochs to be 50. 20% of the training data is left out for model evaluation and model selection. Early stopping is triggered if the challenge metric on the validation set does not grow for 8 epochs. Two typical training processes are depicted in Figure 4. In most cases throughout this paper, when referring to “validation set”, we mean this left-out set from the whole of the public training set, rather than the hidden validation set of the CinC2021 challenge.

4.1. Loss Functions

Since the challenge data is highly unbalanced, having a long tail distribution, we tested 2 loss functions aiming at dealing with such data distribution. Initially, we used weighted binary cross entropy (BCE) as the loss function with weights inversely proportional to the number of records of the classes. After studying other challengers' solutions, we found that the asymmetric loss [28] is more widely adopted. The asymmetric loss is defined as follows

$$ASL = \begin{cases} L_+ := (1-p)^{\gamma_+} \log(p), \\ L_- := (p_m)^{\gamma_-} \log(1-p_m), \end{cases}$$

where $p_m = \max(p-m, 0)$ is the so-called shifted probability, with probability margin m. The loss on one label of one sample is

$$L = -yL_+ - (1-y)L_-$$

By using this asymmetric loss, one is able to emphasize the contribution of positive samples or negative samples by adjusting the ratio of the focusing parameters γ_+ to γ_-.

Typically in our experiments, γ_+ is fixed to be 0, and we set $\gamma_- = 0.2$. We observed augments of challenge metric by 0.03-0.05 on the validation set, which is very significant. More details can be found in Section 7.1. Therefore the asymmetric loss is our final choice of the loss function for training the neural network models.

4.2. Optimizers and Learning Rate Schedulers

Parameters of neural network models are optimized using the AMSGrad variant of the AdamW optimizer (denoted as `adamw_amsgrad`) [29]. For learning rate, which is the most important hyperparameter of an optimizer, we use the `OneCycle` scheduler [30] with maximum learning rate 0.002 and cosine annealing strategy to adjust learning rate during training. It was reported in the deep learning research community that the combination of `adamw_amsgrad` optimizer with `OneCycle` scheduler gives the best performance most of the time. For the purpose of comparative study, we also conduction experiments without any learning rate schedulers starting from an initial learning rate of 0.001, letting the optimizer `adamw_amsgrad` do self-adaptation of the learning rate during training. Corresponding results will be gathered in Section 7.1.

4.3. Model Inference and Model Selection

Model inference is done in a self-adaptive way. More precisely, to make binary predictions from the probability vector, a hard threshold 0.5 is used. If none of the probabilities exceeds 0.5, then the class with the highest probability would be chosen as the binary output, along with other classes that are close enough in probability (the difference ≤ 0.03) with the highest one.

The monitor for model selection is the challenge score computed on the validation set.

5. Clinical Rule Based Detectors

In order to assist the deep learning models, as well as to improve interpretability, we design clinical rules based detectors for several ECG abnormalities. From the authors' previous research experience and experiences in designing industrial ECG auxiliary diagnosis systems, rule-based post-processing is often a helpful supplement to machine learning models. We list a few of them as follows:

1. "LAD" (left axis deviation) and "RAD" (right axis deviation): these two abnormalities are detected by checking the positivity of the QRS complexes in leads I, II, aVF (the "3-lead" method) or in leads I, aVF (the "2-lead" method) as in [31]. More precisely
   - "2-lead" method:
     - "LAD": lead I is positive; lead aVF is negative;
     - "RAD": lead I is negative; lead aVF is positive.
   - "3-lead" method:
     - "LAD": lead I is positive; lead II, aVF are negative;
     - "RAD": lead I is negative; lead II, aVF are positive.

   the "2-lead" method is simpler, but might have false positives.
2. "LQRSV" (low QRS voltage): this abnormality is directly related to the absolute values of the ECGs, hence is distinctive from almost all of the rest ECG abnormalities. The detection is done by checking peak-to-peak amplitudes of the QRS complexes in the limb leads (I, II, III, aVR, aVL, aVF), or in the precordial leads (V1-V6). If the QRS detection were to fail, then peak-to-peak amplitudes are computed as the amplitudes of sliding windows of 0.12 second, which is almost the duration of a QRS complex.

It should be emphasized that the a large proportion of clinical rule based detectors rely heavily on the quality of R peak detection, which is another research topic other than classification for ECG that we do not consider in this work.

Confusion matrices of clinical rule based detectors on 5 typical classes in the CinC2021 challenge database, using the standard 12-lead ECGs, is collected in Table 4

It can be inferred from the clinical rules that a small proportion of ECG arrhythmias can not be reasonably detected under certain combination of reduced leads, resulting in potential drop in model performances, as would be illustrated in Section 6.

class	TP	FP	TN	FN
brady*	18905	5342	54403	306
LAD	6547	13547	57778	1084
RAD	927	3723	73953	353
LQRSV	787	2798	74671	700
PR	671	55	77421	809

Table 4: Confusion matrices of clinical rule based detectors on 5 typical classes in the CinC2021 challenge database. All 12 leads are used. Abbreviations are "brady" for bradycardia, "LAD" for left axis deviation, "RAD" for right axis deviation, "LQRSV" for low QRS voltage, "PR" for pacing rhythm; "TP" for true positive, "FP" for false positive, "TN" for true negative, "FN" for false negative.
* we merge the 2 classes "Brady" and "SB" (sinus bradycardia) in the CinC2021 challenge database into one for applying clinical rule based detectors.

6. Results

In this section, our major findings for the problems raised in the begging of this work are presented.

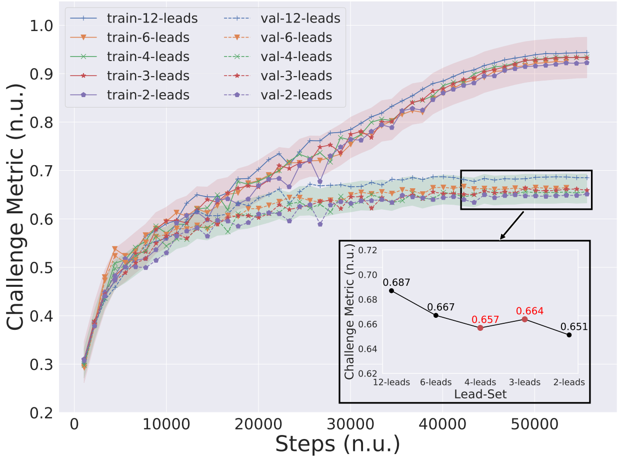


Figure 5: Curves of challenge scores on both the training and validation set from the baseline model tested on the 5 lead-sets.

6.1. Data Redundancy

For studying the problem of data redundancy of the standard 12-lead ECGs, we adopt the baseline model with the CRNN architecture with the following (sequential) components:

- CNN backbone: `ResNet-NC-SE`;
- RNN module: stack of 2 bidirectional LSTM layers with hidden size 192;
- Attention module: SE module with reduction ratio 8;
- MLP head: stack of 2 linear layers with 1024 intermediate features.

Note that in [4], the feature map from the CNN backbone are flattened (which can be called “depth to space”) to feed into a linear layer for final prediction, while we just use a global max pooling layer before the linear layers.

The curves of challenge scores for this set of experiments are plotted in Figure 5. We find that the drop from using all the 12 leads to using only 2 of them for making ECG arrhythmia classifications is very slight (only 0.036 in terms of challenge metric) and is reasonably acceptable. Another interesting phenomenon is that the model performance is higher on the 3-lead-set (I, II, V2) than on the 4-lead-set (I, II, III, V2), although the former is a subset of the latter. This phenomenon has already been observed in our CinC2021 challenge official phase solution, as well as other participants and the challenge organizers. The extra information from lead III seems to be harmful to the neural networks.

6.2. CNN Backbone Effectiveness

To verify the effectiveness of the CNN backbones, we remove extra components from the CRNN framework, leaving only the CNN backbone along with a linear layer for producing the probability vector. The CNN backbones we compare include 3 variants of [4] and 3 variants of TResNet as listed in Table 3, as well as the 3-branch CNN as described in Table 1 and its “lead-wise” variant which we used in the CinC2021 challenge. Their performances in terms of the challenge score on the validation set against the model sizes in terms of the number of trainable parameters is shown in Figure 6. We can see in this figure that the variants of [4] are more effective compared against others while having moderate sizes, especially the ones with bottleneck building block. It should be noted that authors of TResNet [23] replaces all basic building blocks with bottleneck building block in their TResNet V2 in their GitHub repository. This might be another evidence of the superiority of the bottleneck building block over the basic building block.

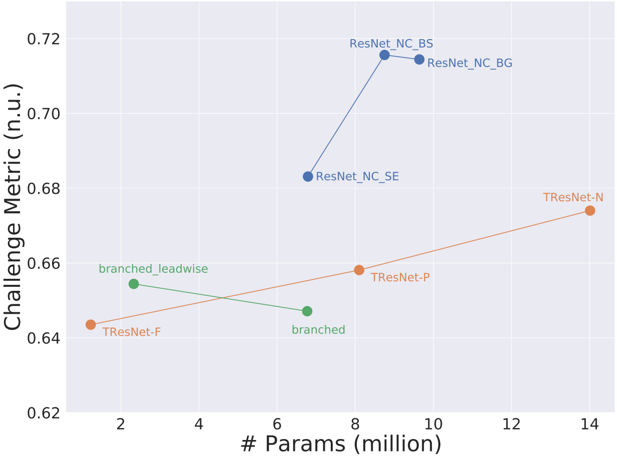


Figure 6: Plot of the performances of the CNN backbones against their sizes in number of trainable parameters.

7. Ablation Study and Beyond

7.1. Effect of Loss Function, Normalization, Augmentations, and More

As is already mentioned in Section 4.1, we find that the asymmetric loss largely improves the performance of neural network models. We are going to show this point in this section. We also conduct various experiments on other techniques including band-pass filtering, normalization, augmentations, learning rate schedulers, etc., aiming at figuring out whether they truly help augmenting model performances or not. This servers as a part of our ablation studies in a broader sense, which we call Ablation Study 0.

We list the choices for the components for training neural networks that are altered in this part of ablation study:

- With or without band-pass filtering;
- Loss functions: asymmetric loss, weighted BCE loss;
- Normalization: z-score, no normalization;
- Augmentations: `mixup`, label smoothing;
- Learning rate scheduler: `OneCycle` scheduler, self-adaptive learning rate by `adamw_amsgrad` with initial value 0.001;

The baseline model for conducting this part of ablation study is the CRNN model used in Section 6.1.

Figure 7 demonstrates the growth of the challenge score on both the training and the validation set. Mini-batch loss against the number of steps is plotted in Figure 8. One can draw the conclusion that the asymmetric loss contributes most to the improvement of the baseline model. The `OneCycle` learning rate scheduler takes the second place but the augment is much less significant. Influences of other techniques are small. The loss curves on the training set with augmentation techniques have larger oscillations, while the loss curve without any augmentation techniques (the green "asymmetric-onecyle" curve with cross markers in Figure 8) decreases more stably. It is surprising that without the band-pass filtering preprocessing procedure, the training loss curve has the largest oscillation while the trained model performs the best on the validation set.

There are other phenomena we want to emphasize but not included in this ablation study. To name a few: performance of neural network models is severely deteriorated if low order band-pass filtered is added into the data preprocessing pipeline, in which case the loss in the passband is non-neglectable. Batch size also affects the model performance, in which case hyperparameters of the learning rate scheduler should be carefully tuned to fit the batch size. This problem has been discussed in literature [32] and in less formal papers [33].

7.2. Effect of Neural Network Complexity

Intuitively, increasing the network complexity helps improving model performances. However, it is not always true, sometimes even the opposite. We designed the following sets of experiments to check the contributions of various component of the whole network to the challenge task.

1. Ablation Study 1: we fix the CNN backbone `ResNet-NC-SE`, and gradually shrink the other parts:
   - LSTM + SE attention layer + 2 linear layers (denoted `lstm-se-2linear`);
   - SE attention layer + 2 linear layers (denoted `se-2linear`);
   - 2 linear layers (denoted `2linear`);
   - 1 linear layer (denoted `1linear`).
2. Ablation Study 2: we change the CNN backbones, and fix the rest part of the network to be LSTM + SE attention

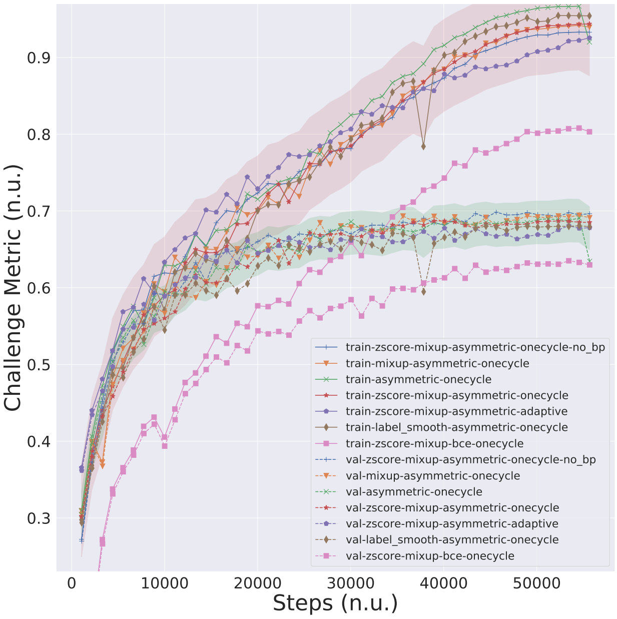


Figure 7: Ablation Study 0: Curves of challenge scores on both the training and validation sets. The techniques used are concatenated by "-" in the legend of the figure. The suffix "no_bp" indicates that the input ECGs are not band-pass filtered before fed into the model.

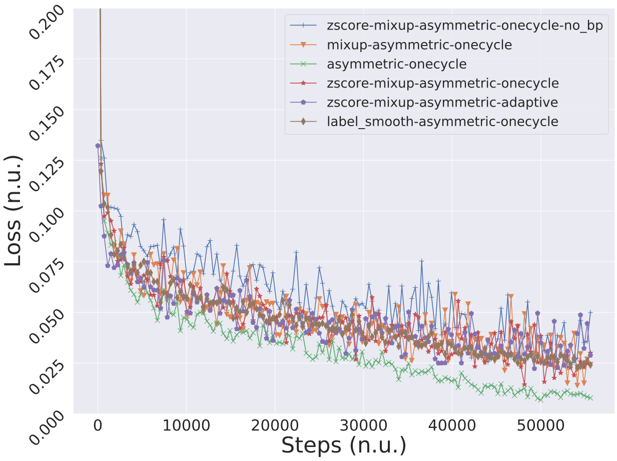


Figure 8: Ablation Study 0: Curves of (mean value of) Mini-batch asymmetric loss on the training set. The experiment using the weighted BCE loss is not included, since it's meaningless to compare it with the rest 4. The curves are smoothed using exponential moving average (EMA) with weight 0.6. Legends have the same meaning as in Figure 8.

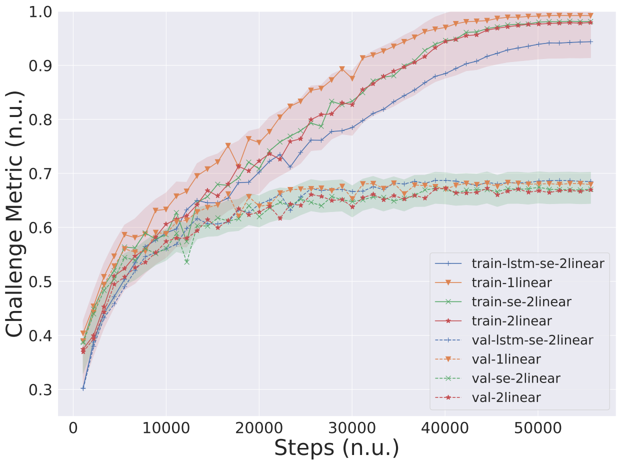


Figure 9: Ablation Study 1: Curves of challenge scores on both the training and validation sets using 12 leads ECGs. Names of neural network components other than the `ResNet-NC-SE` backbone are concatenated by “-” in the legend.

layer + 2 linear layers. The CNN backbones includes the ResNet variants introduced in Section 3.4, as well as the multi-branched CNN introduce in Section 3.2 and its lead-wise variant introduce in Section 3.3.

3. Ablation Study 3: this ablation study has similar setting as in Ablation 2, the difference is that the rest part of the network other than the CNN backbone is changed to the simplest single linear layer.

One can observe in Figure 9 that the 4 combinations listed in Ablation Study 1 have very close performances on the validation set. However, the extra LSTM layer largely suppresses the overfitting on the training set. The other 2 combinations (`se-2linear` and `2linear`) with extra components other than one bare linear layer have lowered performances. Increasing the network complexity does not always result in better performances.

With the help of extra modules (`lstm-se-2linear`), neural networks with different CNN backbones also have very close performances as illustrated in Figure 10. The gap between the challenge scores on the training set and on the validation set is significantly smaller when using the `ResNet-NC-BS` backbone. The bottleneck building block seems to have the effect of regularization to some extent.

When CNN backbones are concatenated with only a bare linear layer for prediction, performances of the networks diverge, as can be observed in Figure 11. Networks completely without any extra structures (inner SE, bottleneck, etc.) degrade dramatically. Other networks with higher complexity overfit very fast on the training set.

To fully validate the findings in Ablation Study 1, we

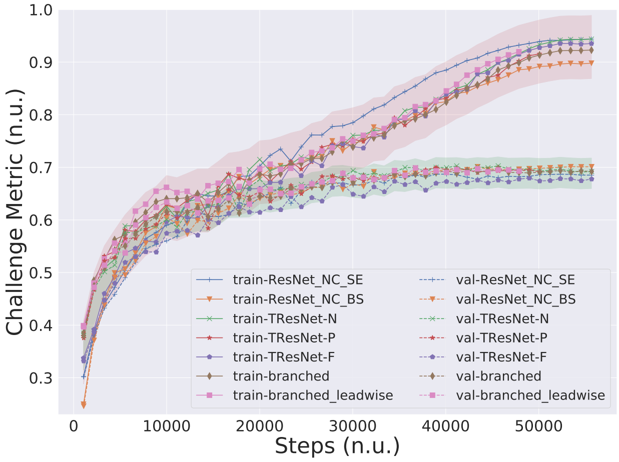


Figure 10: Ablation Study 2: Curves of challenge scores on both the training and validation sets using 12 leads ECGs from neural networks composed of different CNN backbones followed by LSTM module, SE module, and two linear layers.

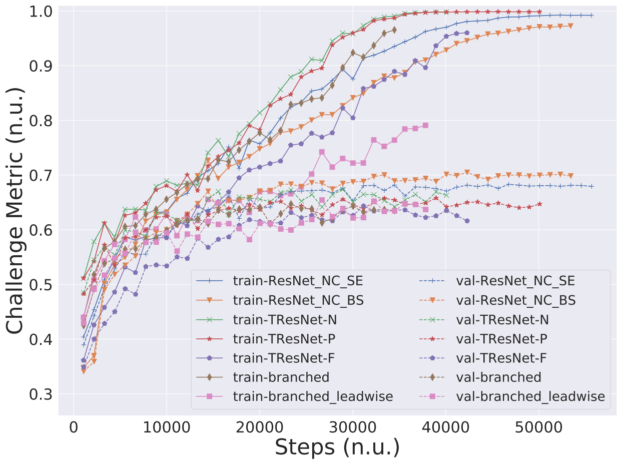


Figure 11: Ablation Study 3: Curves of challenge scores on both the training and validation sets using 12 leads ECGs from neural networks composed of different CNN backbones followed by only one linear layers. It should be noted that the training setups for this set of experiments have slight difference against experiments in Figure 6, hence the performance of `ResNet-NC-BS` slightly drops. But its superiority against others is unchanged.

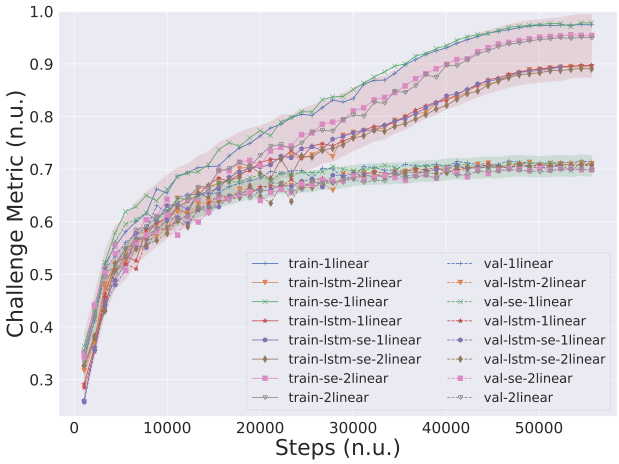


Figure 12: Curves of challenge scores on both the training and validation sets using 12 leads ECGs for a set of CRNN models. The CNN backbone is `ResNet-NC-BS`, the rest components of the models can be read from the legend of this figure.

conduct an additional ablation study using the CNN backbone `ResNet-NC-BS` and with full set of all combinations (totally 8) of modules listed in Ablation Study 1. The results are gathered in Figure 12. The simplest combination, i.e. a backbone `ResNet-NC-BS` with a linear layer performs the best. Our claim that increasing the network complexity does not always result in better performances is verified again.

The overall statistics of all the ablation studies are gathered in Table 5.

8. Discussion and Conclusions

We can conclude from the statistics of the experiments and from the CinC2021 challenge results that the neural network architectures we designed in Section 3 provides effective solution to the problem of detecting a wide range of ECG abnormalities. These neural network models are robust enough such that their performance deterioration on reduced-lead ECGs are neglectable. This gives positive answers to the critical problems raised by the CinC2021 challenge. Furthermore, the “lead-wise” mechanism offers the potential to reduce model size, as well as to reuse model parameters.

Another contribution of this work is that it exhibits the impact on the performance of neural networks from various aspects, for example from the data augmentation techniques through extensive ablation studies. It validates some principles of network design, choices of loss function, etc. This lays solid foundation for further research on related topics in the future.

However, our work still has its limitations and would be further studied and improved in our future work. First of all, the neural architecture search is not so extensive since many other network architectures, especially those can be inherited from computer vision, have not been experimented. Other powerful modules from other application fields of deep learning are not studied as well, for example the transformer encoders[34]. Second, the mechanism of parameter reuse for “lead-wise” CNNs have not been established and implemented. This needs to extract channel-specific weights from the larger weight matrices in the neural networks trained on the standard 12-lead ECGs. An easier alternative method is to fill zeros to the input ECGs in the left-out leads.

The problem of label heterogeneity and insufficiency across the 5 datasets listed in Section 2.1 should also be noted. We observed labels that violates clinical criteria. For example, some “LAD” records violates the “3-lead” method which is more exact than the “2-lead” method mentioned in Section 5. An example is illustrated in Figure 13. A more common situation is that a dataset lacks labels of specific types of ECG abnormality while some records inside this dataset should be diagnosed with such abnormalities. For example, the dataset CPSC [9] have labels related neither to abnormalities of electrical axis (namely “LAD” and “RAD”) nor to abnormalities of amplitudes (e.g. “LQRSV”). However some of its records should be diagnosed with one of such abnormalities, as illustrated in Figure 14 and in Figure 15. Directly training with such data would definitely harm the effectiveness of the models. The question of how to train models in a more proper way under such complicated circumstance is to be studied in our future work.

Acknowledgments

The authors would like to thank professor Deren Han from LMIB, School of Mathematical Sciences, Beihang University and professor Wenjian Yu from the Department of Computer Science and Technology, BNRist, Tsinghua University for generously providing GPU servers to help accomplish this work.

Code Availability

Code, configurations and auxiliary data are all available at https://github.com/DeepPSP/cinc2021

References

[1] Virani SS, Alonso A, Aparicio HJ, Benjamin EJ, Bittencourt MS, Callaway CW, et al. Heart Disease and Stroke Statistics – 2021 Update: a Report from the American Heart Association. Circulation 2021;143(8):e254–e743.

[2] Hannun AY, Rajpurkar P, Haghpanahi M, Tison GH, Bourn C, Turakhia MP, et al. Cardiologist-level Arrhythmia Detec-

CNN backbone	# leads	LSTM	SE	# linear	# params	Best Score	Params Efficiency	Training Speed (sig/s)	Inference Speed (sig/s)
ResNet-NC-SE	12	×	×	1	6.79 M	0.683	10.1	399	2917
	12	×	×	2	7.14 M	0.673	9.42	391	2889
	12	×	✓	2	7.17 M	0.674	9.41	386	2954
	12	✓	✓	2	8.92 M	0.687	7.70	352	2870
ResNet-NC-BS	12	×	×	1	8.75 M	**0.716**	8.18	344	2007
	12	×	×	2	10.1 M	0.700	6.97	338	2026
	12	×	✓	1	9.16 M	0.712	7.77	334	2024
	12	×	✓	2	10.5 M	0.702	6.71	326	2010
	12	✓	×	1	11.9 M	0.710	5.98	313	1966
	12	✓	×	2	12.3 M	0.713	5.80	310	1940
	12	✓	✓	1	11.9 M	0.708	5.94	302	1955
	12	✓	✓	2	12.3 M	0.708	5.74	302	1949
TResNet-N	12	×	×	1	14.0 M	0.674	4.81	244	1395
	12	✓	✓	2	18.4 M	0.703	3.83	182	1139
TResNet-P	12	×	×	1	8.10 M	0.658	5.62	378	2046
	12	✓	✓	2	12.5 M	0.700	8.12	252	1553
TResNet-F	12	×	×	1	**1.23 M**	0.643	**52.3**	373	**3325**
	12	✓	✓	2	4.42 M	0.679	15.4	262	1785
Branched	12	×	×	1	6.77 M	0.647	9.56	**408**	515
	12	✓	✓	2	10.2 M	0.698	6.88	146	362
Branched-LW	12	×	×	1	2.33 M	0.654	28.0	357	536
	12	✓	✓	2	7.46 M	0.699	9.37	134	360

Table 5: Overall statistics of Ablation Study 1, 2, 3 and the additional ablation study. Best scores are computed on the validation set. Params Efficiency is computed as $10^8 \times (\text{Best Score})/(\text{\# Params})$. Training speed is computed under the settings stated in Section 4. Inference speed is computed using one RTX3080 based on a single run on the training set.

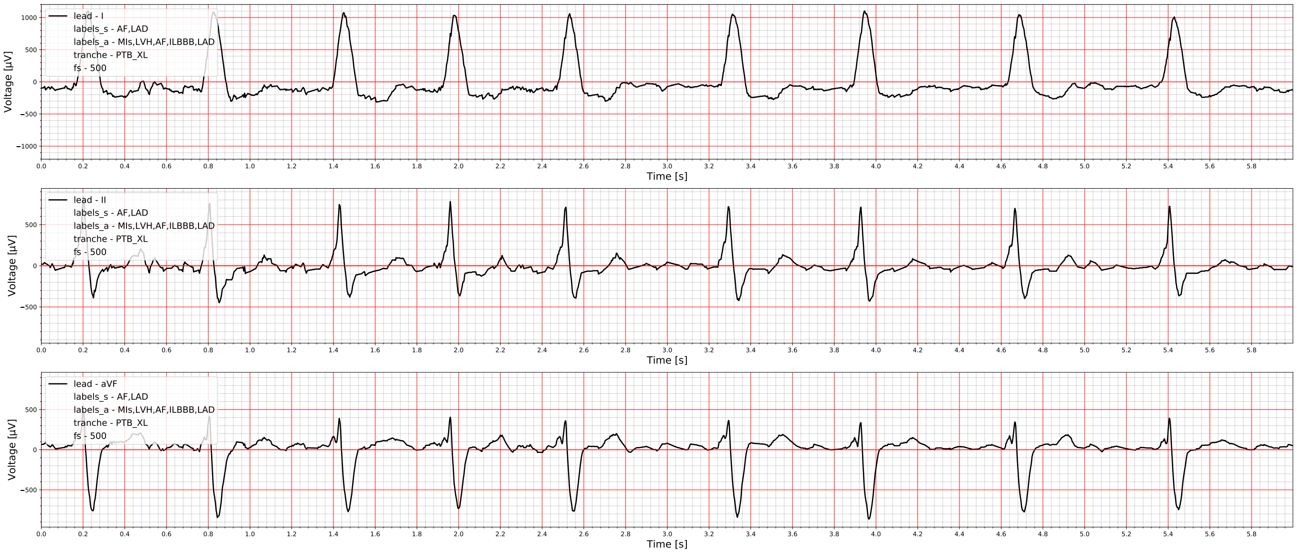


Figure 13: Plot of the first 6 seconds of leads I, II, aVF of the record “HR05631” from the PTB-XL subset. It has “LAD” in its scored label list. Leads I and II are positive, lead aVF is negative. It will be classified as “LAD” by the “2-lead” method detector proposed in Section 5, while classified as non-“LAD” by the “3-lead” method detector.

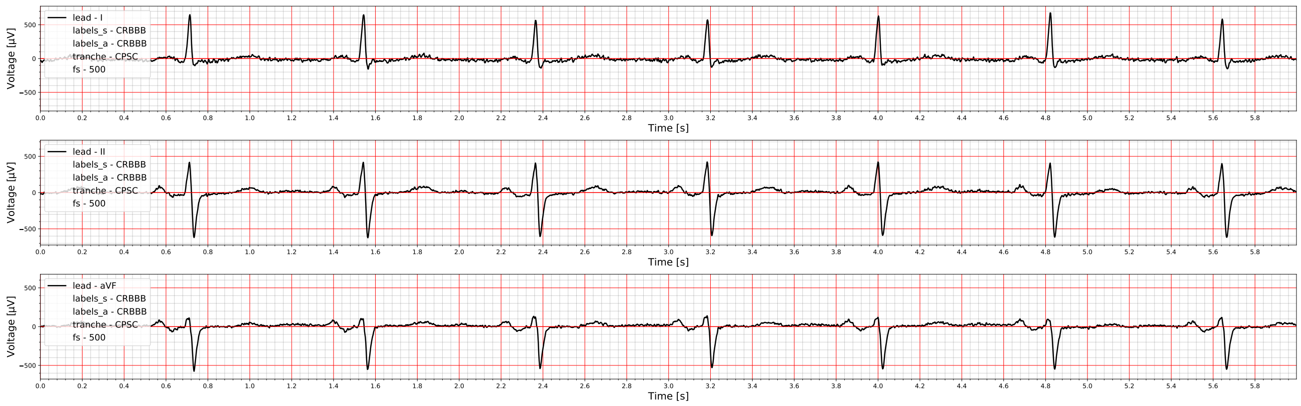


Figure 14: Plot of the first 6 seconds of leads I, II, aVF of the record "A0226" from the CPSC subset. It has label "CRBBB". Lead I is positive, leads II and aVF are negative, hence this record should be classified as "LAD" by the "3-lead" method detector proposed in Section 5. If using the "2-lead" method, this record would definitely be classified as 'LAD". There are many more such examples in the CPSC subset, which contribute many false positives in Table 4.

tion and Classification in Ambulatory Electrocardiograms using a Deep Neural Network. Nature Medicine 2019; 25(1):65.

[3] Yao Q, Wang R, Fan X, Liu J, Li Y. Multi-class Arrhythmia Detection from 12-lead Varied-length ECG Using Attention-based Time-Incremental Convolutional Neural Network. Information Fusion 2020;53:174–182.

[4] Ribeiro AH, Ribeiro MH, Paixão GM, Oliveira DM, Gomes PR, Canazart JA, et al. Automatic Diagnosis of the 12-lead ECG using a Deep Neural Network. Nature Communications 2020;11(1):1–9.

[5] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 2000;101(23):e215–e220.

[6] Perez Alday EA, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, et al. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Physiological Measurement Nov 2020;41. Doi: 10.1088/1361-6579/abc960.

[7] Reyna MA, Sadr N, Perez Alday EA, Gu A, Shah A, Robichaux C, et al. Will Two Do? Varying Dimensions in Electrocardiography: the PhysioNet/Computing in Cardiology Challenge 2021. Computing in Cardiology 2021;48:1–4.

[8] Tihonenko V, Khaustov A, Ivanov S, Rivin A, Yakushenko E. St Petersburg INCART 12-lead Arrhythmia Database. PhysioBank PhysioToolkit and PhysioNet 2008;Doi: 10.13026/C2V88N.

[9] Liu F, Liu C, Zhao L, Zhang X, Wu X, Xu X, et al. An Open Access Database for Evaluating the Algorithms of Electrocardiogram Rhythm and Morphology Abnormality Detection. Journal of Medical Imaging and Health Informatics September 2018;8(7):1368–1373.

[10] Bousseljot R, Kreiseler D, Schnabel A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik 1995;40(S1):317–318.

[11] Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, et al. PTB-XL, a Large Publicly Available Electrocardiography Dataset. Scientific Data 2020;7(1):1–15.

[12] Zheng J, Zhang J, Danioko S, Yao H, Guo H, Rakovski C. A 12-lead Electrocardiogram Database for Arrhythmia Research Covering More Than 10,000 Patients. Scientific Data 2020;7(48):1–8.

[13] Zheng J, Cui H, Struppa D, Zhang J, Yacoub SM, El-Askary H, et al. Optimal Multi-Stage Arrhythmia Classification Approach. Scientific Data 2020;10(2898):1–17.

[14] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. International Conference on Learning Representations 2018;URL https://openreview.net/forum?id=r1Ddp1-Rb.

[15] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; 2818–2826.

[16] Cai W, Hu D. QRS Complex Detection Using Novel Deep Learning Neural Networks. IEEE Access 2020;8:97082–97089.

[17] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation 1997;9(8):1735–1780.

[18] Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020;42(8):2011–2023.

[19] Cao Y, Xu J, Lin S, Wei F, Hu H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). 2019; 1971–1980. Doi: 10.1109/ICCVW.2019.00246.

[20] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected

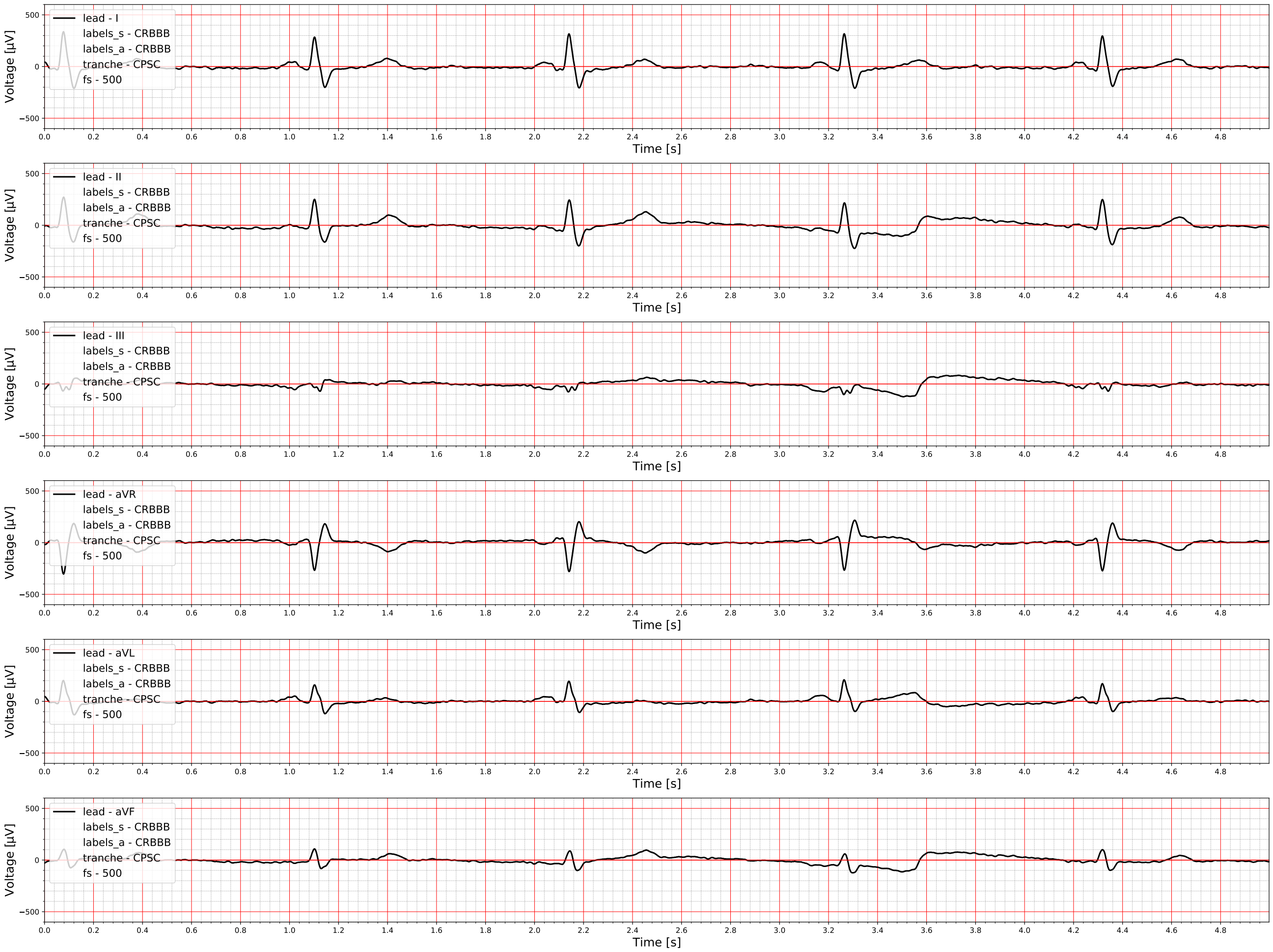


Figure 15: Plot of the first 6 seconds of the limb leads (I, II, III, aVR, aVL, aVF) of the record "A0095" from the CPSC subset. It has label "CRBBB". All its values are < 5mm (0.5mV) in the limb leads, hence should be classified as "LQRSV". Moreover, it should be classified as "LAD" by the "2-lead" method detector proposed in Section 5, although which is a mis-classification.

CRFs. IEEE Transactions on Pattern Analysis Machine Intelligence 2018;40(04):834–848.

[21] Wu Y, He K. Group Normalization. International Journal of Computer Vision 7 2019;128(3):742–755. Doi: 10.1007/s11263-019-01198-w.

[22] He T, Zhang Z, Zhang H, Zhang Z, Xie J, Li M. Bag of Tricks for Image Classification with Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; 558–567.

[23] Ridnik T, Lawen H, Noy A, Ben Baruch E, Sharir G, Friedman I. TResNet: High Performance GPU-Dedicated Architecture. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021; 1400–1409.

[24] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 6 2016; 770–778.

[25] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common Objects in Context. In European conference on computer vision. Springer, 2014; 740–755.

[26] Sandler M, Baccash J, Zhmoginov A, Howard A. Non-Discriminative Data or Weak Model? On the Relative Importance of Data and Model Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019; 0–0.

[27] Zhang R. Making Convolutional Networks Shift-Invariant Again. In International conference on machine learning. PMLR, 2019; 7324–7334.

[28] Ridnik T, Ben-Baruch E, Zamir N, Noy A, Friedman I, Protter M, et al. Asymmetric loss for multi-label classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021; 82–91.

[29] Reddi SJ, Kale S, Kumar S. On the Convergence of Adam and Beyond. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada. OpenReview.net, 2018; .

[30] Smith LN, Topin N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, volume 11006. International Society for Optics and Photonics, 2019; 1100612.

[31] Kashou AH, Basit H, Chhabra L. Electrical Right and Left Axis Deviation. StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing, Jan 2021. Available from: https://www.ncbi.nlm.nih.gov/books/NBK470532/.

[32] Jastrzebski S, Kenton Z, Arpit D, Ballas N, Fischer A, Bengio Y, et al. Width of Minima Reached by Stochastic Gradient Descent is Influenced by Learning Rate to Batch Size Ratio. In International Conference on Artificial Neural Networks. Springer, 2018; 392–402.

[33] Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, et al. Accurate, Large Minibatch SGD: Training Imagenet in 1 Hour. arXiv preprint arXiv170602677 2017; .

[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All You Need. In Advances in Neural Information Processing Systems. 2017; 5998–6008.

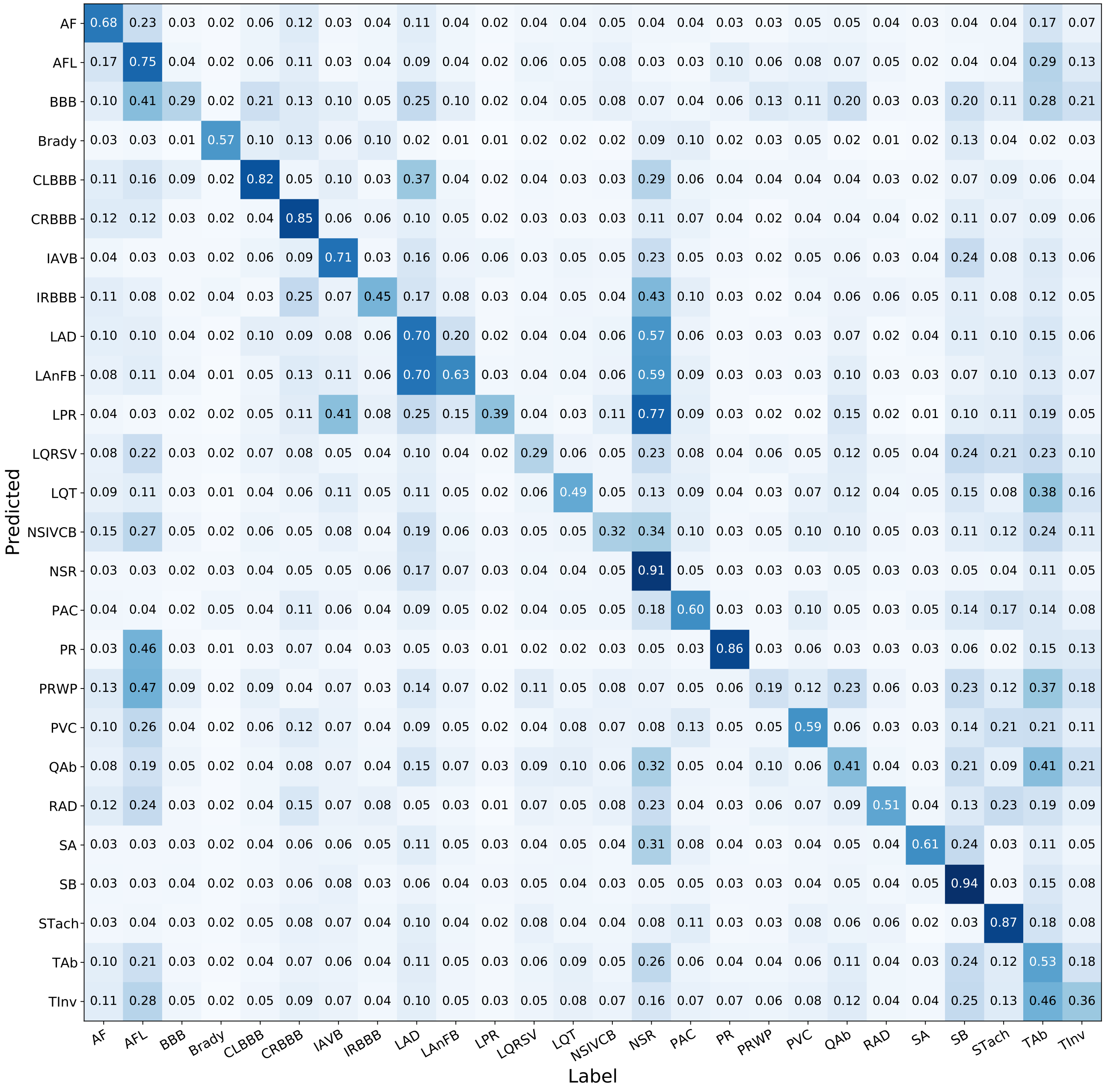
Appendix

A. Probability Matrix of a Typical Neural Network Model

We plot the matrix of predicted probabilities on the validation set for the best neural network model `ResNet-NC-BS` in Figure 16. It should be noted that the data processing pipeline for this model includes the Z-score normalization. We can see from this figure that the model performs particularly bad on 2 classes, namely "LQRSV" (low QRS voltage) and "PR" (poor R wave progression). The former is described in Section 5, clinical diagnostic criteria for the latter is "absence of the normal increase in size of the R wave in the precordial leads when advancing from lead V1 to V6". These 2 classes are exactly the only 2 among the 26 scored classes in the CinC2021 challenge database that are directly related to the amplitudes of the ECGs. The normalization operation destroys such characteristics. With the help of clinical rule based detectors, such performance deterioration of the neural network models could be alleviated. (ref. the "TP" column and "FN" column of the Table 4.) Clinical rule based detectors are truly helpful assists to neural network models.

Address for correspondence:

Hao Wen
E301, Beihang University, Changping District, Beijing, China
wenh06@buaa.edu.cn, wenh06@gmail.com



	AF	AFL	BBB	Brady	CLBBB	CRBBB	IAVB	IRBBB	LAD	LAnFB	LPR	LQRSV	LQT	NSIVCB	NSR	PAC	PR	PRWP	PVC	QAb	RAD	SA	SB	STach	TAb	TInv
AF	0.68	0.23	0.03	0.02	0.06	0.12	0.03	0.04	0.11	0.04	0.02	0.04	0.04	0.05	0.04	0.04	0.03	0.03	0.05	0.05	0.04	0.03	0.04	0.04	0.17	0.07
AFL	0.17	0.75	0.04	0.02	0.06	0.11	0.03	0.04	0.09	0.04	0.02	0.06	0.05	0.08	0.03	0.03	0.10	0.06	0.08	0.07	0.05	0.02	0.04	0.04	0.29	0.13
BBB	0.10	0.41	0.29	0.02	0.21	0.13	0.10	0.05	0.25	0.10	0.02	0.04	0.05	0.08	0.07	0.04	0.06	0.13	0.11	0.20	0.03	0.03	0.20	0.11	0.28	0.21
Brady	0.03	0.03	0.01	0.57	0.10	0.13	0.06	0.10	0.02	0.01	0.01	0.02	0.02	0.02	0.09	0.10	0.02	0.03	0.05	0.02	0.01	0.02	0.13	0.04	0.02	0.03
CLBBB	0.11	0.16	0.09	0.02	0.82	0.05	0.10	0.03	0.37	0.04	0.02	0.04	0.03	0.03	0.29	0.06	0.04	0.04	0.04	0.04	0.03	0.02	0.07	0.09	0.06	0.04
CRBBB	0.12	0.12	0.03	0.02	0.04	0.85	0.06	0.06	0.10	0.05	0.02	0.03	0.03	0.03	0.11	0.07	0.04	0.02	0.04	0.04	0.04	0.02	0.11	0.07	0.09	0.06
IAVB	0.04	0.03	0.03	0.02	0.06	0.09	0.71	0.03	0.16	0.06	0.06	0.03	0.05	0.05	0.23	0.05	0.03	0.02	0.05	0.06	0.03	0.04	0.24	0.08	0.13	0.06
IRBBB	0.11	0.08	0.02	0.04	0.03	0.25	0.07	0.45	0.17	0.08	0.03	0.04	0.05	0.04	0.43	0.10	0.03	0.02	0.04	0.06	0.06	0.05	0.11	0.08	0.12	0.05
LAD	0.10	0.10	0.04	0.02	0.10	0.09	0.08	0.06	0.70	0.20	0.02	0.04	0.04	0.06	0.57	0.06	0.03	0.03	0.03	0.07	0.02	0.04	0.11	0.10	0.15	0.06
LAnFB	0.08	0.11	0.04	0.01	0.05	0.13	0.11	0.06	0.70	0.63	0.03	0.04	0.04	0.06	0.59	0.09	0.03	0.03	0.03	0.10	0.03	0.03	0.07	0.10	0.13	0.07
LPR	0.04	0.03	0.02	0.02	0.05	0.11	0.41	0.08	0.25	0.15	0.39	0.04	0.03	0.11	0.77	0.09	0.03	0.02	0.02	0.15	0.02	0.01	0.10	0.11	0.19	0.05
LQRSV	0.08	0.22	0.03	0.02	0.07	0.08	0.05	0.04	0.10	0.04	0.02	0.29	0.06	0.05	0.23	0.08	0.04	0.06	0.05	0.12	0.05	0.04	0.24	0.21	0.23	0.10
LQT	0.09	0.11	0.03	0.01	0.04	0.06	0.11	0.05	0.11	0.05	0.02	0.06	0.49	0.05	0.13	0.09	0.04	0.03	0.07	0.12	0.04	0.05	0.15	0.08	0.38	0.16
NSIVCB	0.15	0.27	0.05	0.02	0.06	0.05	0.08	0.04	0.19	0.06	0.03	0.05	0.05	0.32	0.34	0.10	0.03	0.05	0.10	0.10	0.05	0.03	0.11	0.12	0.24	0.11
NSR	0.03	0.03	0.02	0.03	0.04	0.05	0.05	0.06	0.17	0.07	0.03	0.04	0.04	0.05	0.91	0.05	0.03	0.03	0.03	0.05	0.03	0.03	0.05	0.04	0.11	0.05
PAC	0.04	0.04	0.02	0.05	0.04	0.11	0.06	0.04	0.09	0.05	0.02	0.04	0.05	0.05	0.18	0.60	0.03	0.03	0.10	0.05	0.03	0.05	0.14	0.17	0.14	0.08
PR	0.03	0.46	0.03	0.01	0.03	0.07	0.04	0.03	0.05	0.03	0.01	0.02	0.02	0.03	0.05	0.03	0.86	0.03	0.06	0.03	0.03	0.03	0.06	0.02	0.15	0.13
PRWP	0.13	0.47	0.09	0.02	0.09	0.04	0.07	0.03	0.14	0.07	0.02	0.11	0.05	0.08	0.07	0.05	0.06	0.19	0.12	0.23	0.06	0.03	0.23	0.12	0.37	0.18
PVC	0.10	0.26	0.04	0.02	0.06	0.12	0.07	0.04	0.09	0.05	0.02	0.04	0.08	0.07	0.08	0.13	0.05	0.05	0.59	0.06	0.03	0.03	0.14	0.21	0.21	0.11
QAb	0.08	0.19	0.05	0.02	0.04	0.08	0.07	0.04	0.15	0.07	0.03	0.09	0.10	0.06	0.32	0.05	0.04	0.10	0.06	0.41	0.04	0.03	0.21	0.09	0.41	0.21
RAD	0.12	0.24	0.03	0.02	0.04	0.15	0.07	0.08	0.05	0.03	0.01	0.07	0.05	0.08	0.23	0.04	0.03	0.06	0.07	0.09	0.51	0.04	0.13	0.23	0.19	0.09
SA	0.03	0.03	0.03	0.02	0.04	0.06	0.06	0.05	0.11	0.05	0.03	0.04	0.05	0.04	0.31	0.08	0.04	0.03	0.04	0.05	0.04	0.61	0.24	0.03	0.11	0.05
SB	0.03	0.03	0.04	0.02	0.03	0.06	0.08	0.03	0.06	0.04	0.03	0.05	0.04	0.03	0.05	0.05	0.03	0.03	0.04	0.05	0.04	0.05	0.94	0.03	0.15	0.08
STach	0.03	0.04	0.03	0.02	0.05	0.08	0.07	0.04	0.10	0.04	0.02	0.08	0.04	0.04	0.08	0.11	0.03	0.03	0.08	0.06	0.06	0.02	0.03	0.87	0.18	0.08
TAb	0.10	0.21	0.03	0.02	0.04	0.07	0.06	0.04	0.11	0.05	0.03	0.06	0.09	0.05	0.26	0.06	0.04	0.04	0.06	0.11	0.04	0.03	0.24	0.12	0.53	0.18
TInv	0.11	0.28	0.05	0.02	0.05	0.09	0.07	0.04	0.10	0.05	0.03	0.05	0.08	0.07	0.16	0.07	0.07	0.06	0.08	0.12	0.04	0.04	0.25	0.13	0.46	0.36

Figure 16: The matrix of predicted probabilities on the validation set for the best neural network model `ResNet-NC-BS`