Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Domain knowledge enhanced deep learning for electrocardiogram arrhythmia classification[*]

Jie SUN
School of Cyber Science and Engineering, Ningbo University of Technology, Ningbo 315211, China
E-mail: sunjie@nbut.edu.cn

Received Nov. 3, 2021; Revision accepted July 21, 2022; Crosschecked Aug. 16, 2022

Abstract: Deep learning provides an effective way for automatic classification of cardiac arrhythmias, but in clinical decision-making, pure data-driven methods working as black-boxes may lead to unsatisfactory results. A promising solution is combining domain knowledge with deep learning. This paper develops a flexible and extensible framework for integrating domain knowledge with a deep neural network. The model consists of a deep neural network to capture the statistical pattern between input data and the ground-truth label, and a knowledge module to guarantee consistency with the domain knowledge. These two components are trained interactively to bring the best of both worlds. The experiments show that the domain knowledge is valuable in refining the neural network prediction and thus improves accuracy.

Key words: Domain knowledge; Cardiac arrhythmia; Electrocardiogram (ECG); Clinical decision-making
https://doi.org/10.1631/FITEE.2100519 **CLC number:** TP391.5

1 Introduction

In recent years, deep learning technology has provided a new and effective example for making clinical decisions from pathophysiologic data (National Center for Cardiovascular Diseases, 2019). Some works have achieved better performance than a human specialist (Hannun et al., 2019). These successful models are all data-based learning methods; that is, the models take raw electrocardiogram (ECG) data as input, extract features, and output a prediction based on the input data. However, pure data-driven methods may lead to unsatisfactory results due to an unbalanced, incomplete, or biased dataset, and may not meet the constraints prescribed by natural law. A promising solution is integrating domain knowledge in the neural network pipeline to correct the deviation.

In this paper, we propose a general framework to address the questions in ECG arrhythmia classification, including: (1) How to represent the clinical knowledge so that it can be injected into the deep learning architecture? (2) How can domain knowledge affect the deep neural network (DNN) learning process, when the learning is based on gradient descent and back propagation? (3) Does the integration really improve or reduce the performance of the DNN? And how?

2 Related works

In recent years, the DNN model has been applied in the diagnosis of different cardiac diseases, such as heart arrhythmias (Acharya et al., 2019; Baloglu et al., 2019). Although DNNs have had significant success, they still have limitations in specific tasks because they are purely data-driven and are highly dependent on the training data. A solution is to integrate

[*] Project supported by the Scientific Research Fund of Zhejiang Provincial Education Department (No. Y201839519) and the Ningbo Natural Science Foundation (No. 2019A610087)
ORCID: Jie SUN, https://orcid.org/0000-0003-2996-7613

prior knowledge in the training process, and a variety of approaches have been proposed.

2.1 Integrating knowledge with data before feeding them into the DNN model

Domain knowledge can be applied to select appropriate data before they are fed into the DNN model. There are 12 leads in a conventional ECG. The six leads I, II, III, aVR, aVL, and aVF are limb leads, and the six other leads V1, V2, V3, V4, V5, and V6 are precordial leads (Surawicz and Knilans, 2008). Some leads have more pathological value for detection of a particular disease; for example, leads V2, V3, V5, and aVL are more sensitive and valuable in detecting myocardial infarction, and thus the related leads are selected as input instead of all 12 leads (Liu WH et al., 2018).

Domain knowledge can also be applied to analyze the inherent correlation of the input data. A classification model called MBCRNet designs three branches and considers synchronization and orthogonality of multiple leads (Chen B et al., 2018) to explore the different features. The average accuracy is 87.04% and the sensitivity is 89.93%.

2.2 Integrating knowledge after the DNN model makes a prediction

Domain knowledge can be integrated with the DNN by decision fusion methods. The DNN makes a prediction and the clinical knowledge model (represented as diagnosis rules) performs inference separately, and the two results are fused to obtain the final decision (Jin and Dong, 2017).

Many works leveraged domain knowledge to refine the prediction result of a DNN model, which is called post-processing in some literature. Zhou et al. (2017) used ensemble classifiers to divide the ECG records into two categories, premature ventricular contraction (PVC) and non-PVC, and then rule-based inference was performed for each category to further refine the prediction result. Singstad and Tronstad (2020) individually classified 27 cardiac abnormalities with the deep learning model and rule-based algorithm. If there was inconsistency between the two results, the DNN classification result was rewritten by the rule-based algorithm. Parvaneh et al. (2018) applied DenseNet to classify the ECG record into four categories. In view of the high misclassification between the categories “normal sinus rhythm (NSR)” and “other rhythm (O),” once the absolute difference between the predicted probabilities of the two categories was less than a heuristic threshold (0.4 in the paper), a binary classification will start working to make the final decision.

2.3 Integrating knowledge with the DNN model in parallel

A variety of methods have been proposed to integrate knowledge with the DNN model and simultaneously perform training. This paper focuses on the use of logic, more specifically, first-order logic (FOL), to represent domain knowledge.

Rule distillation has been proposed to refine the knowledge represented by FOL rules for the DNN model, where the rules will force the DNN model to simulate the prediction of the rules during training through posterior regularization (Hu et al., 2016).

Logic is not differentiable, so many methods integrate logic rules as constraints or regularization terms of the DNN model, and perform relaxation to make them amenable to gradient-based learning. Semantic based regularization (SBR) represents the logic as a regularization term in the loss function to provide a penalty when the DNN model prediction violates the knowledge (Diligenti et al., 2017). Probabilistic soft logic (PSL) consists of a set of FOL rules and the satisfaction distance of the grounded rules is added to the loss function as a regularization term (Kimmig et al., 2012). Abductive learning is a framework that unifies machine learning and logical reasoning (Dai et al., 2019). In each training epoch, the conventional neural network is used to produce primitive logic facts, called pseudo-labels, and logical reasoning is used to revise incorrect pseudo-labels based on the domain knowledge. The revised labels are used to re-train the neural network in the next epoch.

We prefer this method because the classification model can learn from the data and the domain knowledge jointly. The structural knowledge represented with FOL rules can be integrated into the neural network without changing the DNN model’s training process. Our method applies logic rules to represent domain knowledge, but the weight of each rule is not manually specified and will be regulated and optimized jointly with DNN weights during the learning process. Thus, the knowledge specification will also adapt to the meaningful data.

3 Methods

In this paper, we propose a generalized framework that enables integrated learning of the DNN and domain knowledge. The architecture is composed of three modules (Fig. 1): a baseline DNN classifier, a knowledge inference module, and a joint learning module. The DNN is an arbitrary neural network that takes a preprocessed signal as input and produces the probability of the category to which the input belongs. The knowledge inference module comprises a knowledge base and a rule-grounding, matching, and scoring (GMS) module. The outputs of the DNN model and the knowledge inference module are n-dimensional vectors, where n is the number of categories. The joint learning module will train the DNN model and knowledge inference module with backward propagation.

3.1 Problem setting

The DNN classifier can be formalized as F_c: $X \to \boldsymbol{Y}$, where X is the preprocessed data and $\boldsymbol{Y} \in \mathbb{R}^n$ is the output space. For the training data $\{(x_i, y_i)\}_{i=1}^n$, the output of the classifier is the probability $p_\theta(y_i|x_i)$ that input x_i belongs to category y_i, and θ denotes the parameter of the neural network. The knowledge inference module can be formalized as F_k: $\hat{X} \times \boldsymbol{Y} \to C$, $C \in \mathbb{R}^+$, where $\hat{X}$ is the raw data without preprocessing, and C is the degree to which that input data matches the label. Input data $\hat{X}$ is different from the preprocessed data X in that it is not chopped or padded into segments of fixed length to make it available for DNN processing, which will cause valuable information to be lost with the abandoned segments.

The objective of the framework is to train the neural network under constraints, to simultaneously minimize the classification mismatch and penalize the violation of the knowledge base. The cost function can be represented as

$$\mathcal{L} = \mathcal{L}_C + \lambda \mathcal{L}_K. \tag{1}$$

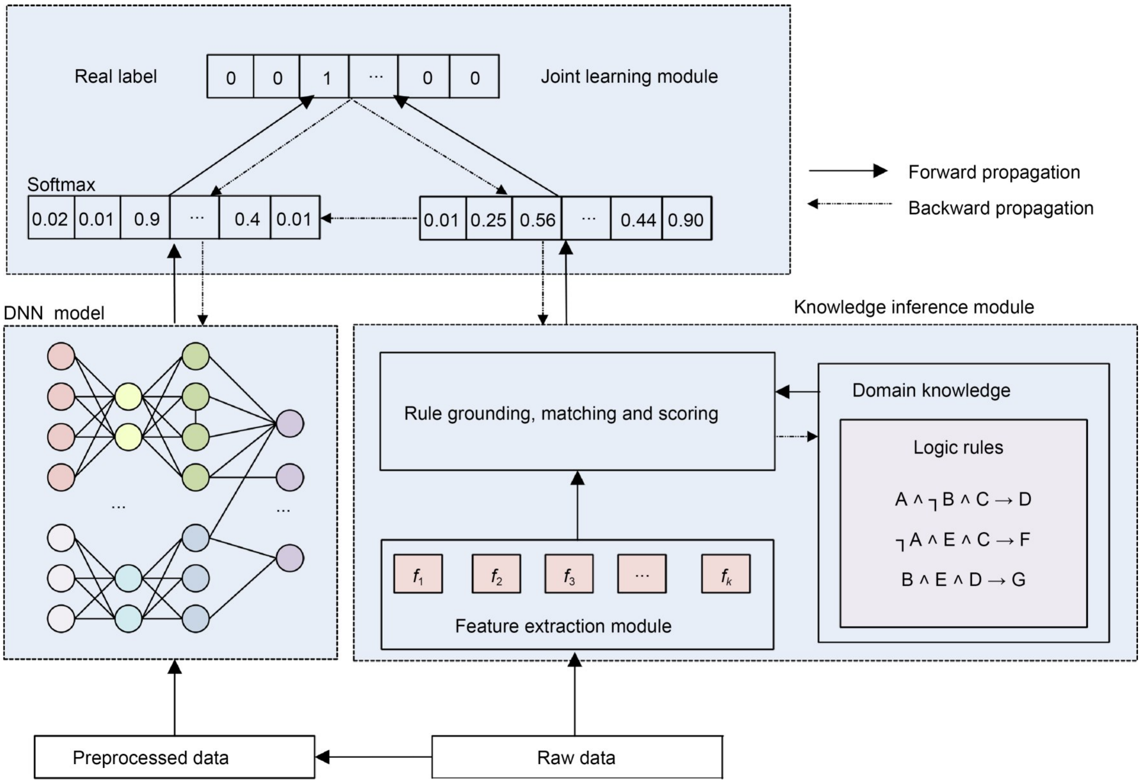

Fig. 1 Architecture of the proposed method

$\mathcal{L}_{\mathrm{C}}$ is used to force the sample to fit the real label, and $\mathcal{L}_{\mathrm{K}}$ is used to penalize the violation between the two outputs of the two modules. λ is a hyperparameter to trade-off between the knowledge inference and deep learning model.

$$\mathcal{L}_{\mathrm{C}} = \frac{1}{N}\sum_{i=1}^{N}\Big(l\big(y_i, p_\theta(y_i|x_i)\big)\Big), \tag{2}$$

where l is the cross-entropy loss function.

$\mathcal{L}_{\mathrm{K}}$ is measured with Kullback-Leibler divergence (Sankaran et al., 2016) in each training iteration:

$$\mathcal{L}_{\mathrm{K}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{KL}\Big(p_{\mathrm{k}}(\hat{x}_i)\,\|\,p_\theta(y_i|x_i)\Big), \tag{3}$$

where $p_{\mathrm{k}}(\hat{x}_i)$ is the knowledge inference module soft prediction, detailed in Section 3.2.2.

3.2 Domain knowledge inference module

3.2.1 Presentation of knowledge

We use fuzzy logic rules to represent the domain knowledge. An atom is a tuple in the form $p(x_1, x_2, \cdots, x_m)$, where $p\in\mathcal{P}$, a given set of base predicates, and x_i is either a variable or a constant. A predicate p is a relation defined by a unique feature extracted as the attributes of an object according to the domain knowledge, such as the permitted value range of the feature. A rule r is a Horn clause of disjunctive predicates with one term in the conclusion part, and each rule is associated with a weight η_r to present the empirically preconfigured confidence of the rule, which can be initialized as 0 and should be updated and learned during training.

$$\eta_r: P_1 \wedge P_2 \cdots \wedge P_m \rightarrow H_r. \tag{4}$$

The rules are stored in the knowledge base. When the training data is input, the features are extracted and the corresponding predicates are grounded. A grounded predicate is the instantiation of all the variables x_i. The set of grounded predicates is also called the Herbrand base, denoted as $\mathcal{G}$. A rule is grounded by grounding all the predicates of the rule iteratively.

For the training data $\{(x_i, y_i)\}_{i=1}^{n}$, the knowledge inference module is formalized as finding a function to compute a real value that represents the satisfaction degree of the grounded rule. Łukasiewicz's t-norm (Klir and Yuan, 1995) is used to define the truth value of basic logical operations, including logical conjunction (∧), disjunction (∨), and negation (¬). The soft truth value is defined as in Table 1.

Table 1 The soft truth computation of Łukasiewicz's *t*-norm

Clause	Soft truth value
$\underset{i}{\wedge}\, p_i$	$\max\left(\sum_i p_i - \lvert P\rvert + 1, 0\right)$
$\underset{i}{\vee}\, p_i$	$\min\left(\sum_i p_i, 1\right)$
$\neg p$	$1 - p$

3.2.2 GMS module

The data input into the knowledge inference module is a complete signal without segmentation or dropout to compensate for the information lost in the preprocessing. Specific features will be extracted from the raw data. The GMS module will use the features to iteratively ground the variables in the rules, determine the satisfied rules, and compute the score of the satisfied rules. The mapping from features to atoms is called an interpretation I. The process is described as follows:

1. Atom translation: When the training data is input, the features are extracted and the corresponding predicates are grounded.
2. Predicate translation: There is a given set $\mathcal{P}$ of base predicates determined according to the domain knowledge, and the predicates are defined as $p(x_1, x_2, \cdots, x_m)$. The predicates are grounded as p or its negation $\neg p$.
3. Proposition translation: The proposition is translated into a combination of predicates with logical operator conjunction (∧) and disjunction (∨).
4. Rule translation: For a rule $r_{\text{body}} \rightarrow r_{\text{head}}$, the soft truth of the antecedent and consequent of the rule are computed as $I(r_{\text{body}})$ and $I(r_{\text{head}})$, respectively, according to Table 1, and the distance under interpretation I to satisfy the rule is defined as $d_r(I) = \max(I(r_{\text{body}}) - I(r_{\text{head}}), 0)$.

Given the grounded atoms, the GMS module derives a distribution over possible interpretations, and the probability density function is defined as

$$P(I)=\frac{1}{Z}\exp\left[-\sum\nolimits_{r\in R}\eta_r d_r(I)\right], \tag{5}$$

where $Z=\int\exp\left[-\sum\nolimits_{r\in R}\eta_r d_r(I)\right]$ is a normalization constant and $r\in R$ is the grounded rule.

We aim to minimize the distance to rule satisfaction for each instance. We compute the distance with the GMS module and find the minimum of all possible rule grounding results.

3.3 Joint learning of the two modules

The loss function $\mathcal{L}$ can be solved if it is convex. By relaxing the logic rules using Łukasiewicz's *t*-norm and limiting the rules as a Horn definite clause, the convexity of $\mathcal{L}_{\mathrm{K}}$ is guaranteed and the loss function can be optimized with the GMS method. Details of the convexity proof can be found in Giannini et al. (2019).

Let θ denote the parameter of the neural network. The gradient of $\mathcal{L}$ with respect to (w.r.t.) θ can be computed as

$$\frac{\partial\mathcal{L}}{\partial\theta}=\frac{\partial(\mathcal{L}_{\mathrm{C}}+\lambda\,\mathcal{L}_{\mathrm{K}})}{\partial\theta}=\frac{\partial\mathcal{L}_{\mathrm{C}}}{\partial\theta}+\lambda\frac{\partial\mathcal{L}_{\mathrm{K}}}{\partial\theta},$$

where

$$\begin{aligned}\frac{\partial\mathcal{L}_{\mathrm{K}}}{\partial\theta}&=\frac{\partial\left(\mathrm{KL}\left(p_{\mathrm{k}}(\hat{x}_i)\,\|\,p_\theta(y_i|x_i)\right)\right)}{\partial\theta}\\&=\sum_{i=1}^{N}\frac{\partial\left(p_{\mathrm{k}}(\hat{x}_i)\ln\dfrac{p_{\mathrm{k}}(\hat{x}_i)}{p_\theta(y_i|x_i)}\right)}{\partial\theta}\\&=-\sum_{i=1}^{N}\left(p_{\mathrm{k}}(\hat{x}_i)\frac{\partial\ln p_\theta(y_i|x_i)}{\partial\theta}\right),\end{aligned}$$

$$\frac{\partial\mathcal{L}}{\partial\theta}=-\sum_{i=1}^{N}\left(\left(y_i+p_{\mathrm{k}}(\hat{x}_i)\right)\nabla p_\theta(y_i|x_i)\right), \tag{6}$$

and ∇p_θ can be computed using the usual neural network backpropagation.

Let η denote the weight of the logic rules. The gradient of $\mathcal{L}$ w.r.t. η can be computed as

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\eta}&=\frac{\partial(\mathcal{L}_{\mathrm{C}}+\lambda\,\mathcal{L}_{\mathrm{K}})}{\partial\eta}\\&=\lambda\frac{\partial\mathcal{L}_{\mathrm{K}}}{\partial\eta}\\&=-\lambda\sum\nolimits_{r\in R}d_r(I)+\mathbb{E}\left[\sum\nolimits_{r\in R}d_r(I)\right].\end{aligned} \tag{7}$$

4 Experiments

This section provides a concrete instance of our general framework in the task of ECG arrhythmia classification. We test the method in detection of eight arrhythmias against normal records from 12-lead ECG signals. The arrhythmias include atrial fibrillation (AF), first-degree atrioventricular block (I-AVB), left bundle branch block (LBBB), right bundle branch block (RBBB), premature atrial contraction (PAC), PVC, ST-segment depression (STD), and ST-segment elevation (STE).

4.1 DNN model

Fig. 2 illustrates the baseline neural network architecture. The input signal in the form of 12×5000 is fed into the first convolutional block, followed by eight convolutional blocks with residual connection and a classification layer. The convolutional blocks have the same structure except for the first and last.

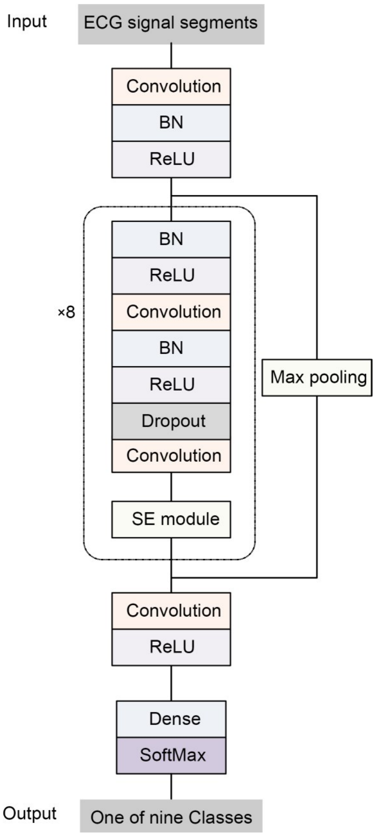

Fig. 2 The DNN model to detect eight arrhythmias against normal records from 12-lead ECG signals

The first convolutional block consists of a one-dimensional convolutional (1D Conv) layer, a batch normalization (BN) layer, and a rectified linear unit (ReLU) layer. BN is the operation ensuring that the dataset has zero mean and unit variances to minimize the impact of an internal covariate shift (Ioffe and Szegedy, 2015), which is the phenomenon that the input distribution of each layer will change with the parameters of the previous layer in the training phase. BN transformation can be added to a network to manipulate any activation and enable a higher learning rate.

For the next eight blocks, each block consists of two convolutional layers. The filter sizes of all the convolutional layers are 16 and the number of filters is 32×2^k, where k starts at 0 and increases by 1 for every four blocks. According to the pre-activation block design, we apply a BN and an ReLU layer before each convolutional layer. We apply residual connection by adding a shortcut connection between two consecutive convolutional blocks. The outputs are added to the outputs of the skipped block. Max pooling is an operation that computes the maximum value of a particular feature and reduces the dimensionality of the output features significantly while enabling a translation invariant of the features. We use max pooling of size 2 and stride 2 in the residual connection to guarantee that the input and output feature maps have the same dimensionality.

The last convolution layer is used to integrate the feature vectors produced. The output of the last convolutional block is fed into a SoftMax regression layer, which corresponds to the probability distribution of the label to which the input ECG segment belongs. A fully connected (FC) layer contains nine cells corresponding to the nine categories.

The squeeze-and-excitation (SE) module is applied to refine the channel-wise feature maps. As shown in Fig. 3, the SE module consists of a global average pooling (GAP) layer, and two FC layers, each with different activation functions. Given the input feature vector as $\boldsymbol{X}$, the GAP layer will squeeze the global spatial information into a channel descriptor to capture channel-wise dependencies. The SE module will produce a scalar s to represent the importance of the channel in Eq. (8), where δ refers to the ReLU function and σ refers to the Sigmoid function.

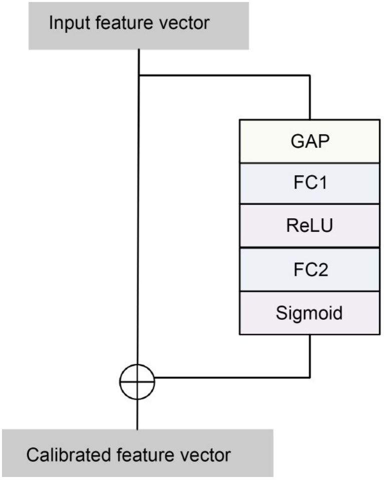

Fig. 3 The squeeze-and-excitation (SE) module

The refined feature vector is shown in Eq. (9), where $s \cdot \boldsymbol{X}$ refers to the channel-wise multiplication between feature vector $\boldsymbol{X}$ and scalar s.

$$s = \sigma(W_2\delta(W_1\mathrm{GAP}(\boldsymbol{X}))), \tag{8}$$

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + s\cdot\boldsymbol{X}. \tag{9}$$

4.2 Domain knowledge

Domain knowledge is used in ECG arrhythmia detection to explore characteristics and improve the classification performance. Knowledge-based rules are aligned with diagnosis criteria according to the cardiologist’s experience and carry clinical meanings.

As shown in Fig. 4, one cardiac cycle in an ECG signal consists of the P-QRS-T waves. The P wave represents atrial depolarization, the QRS complex represents ventricular depolarization, and the ST segment and T wave represent ventricular repolarization (Goldberger et al., 2017). Considering that the symptoms of arrhythmias are different in each lead, the diagnostic rules of cardiac arrhythmia are extracted based on prior knowledge and clinical experience.

When the sinus rhythm is normal, the P wave of lead II is always positive, the P wave of lead aVR is always negative, and the heart rate is between 60 bpm and 100 bpm.

The diagnostic for BBB is performed mainly in a widened QRS complex greater than 0.12 s. RBBB will result in the right ventricle depolarizing after the left ventricle, which can be reflected by leads I, V6, and V1 (indicating the slow depolarization of the

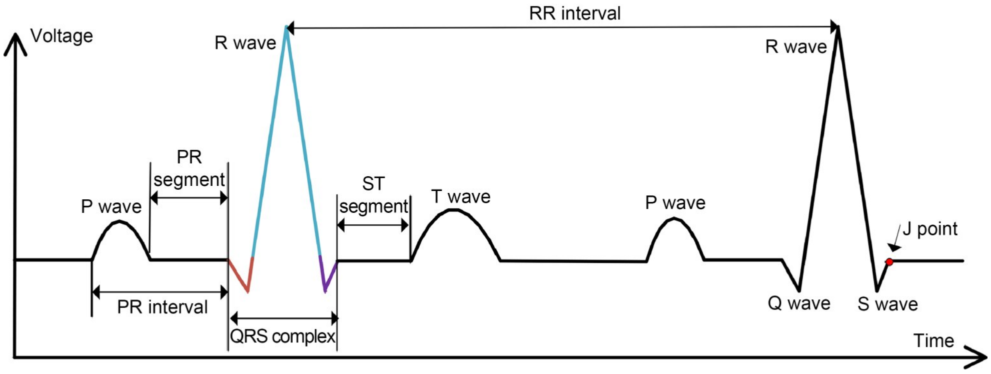

Fig. 4 The cardiac cycle in an ECG signal

The Q, R, and S waves in a QRS complex are plotted in different colors. References to color refer to the online version of this figure

right ventricle in a left-to-right direction). Associated features of diagnostic criteria for RBBB include a wide slurred S wave in leads V5 and V6, ST segment depression, and T wave inversion in lead V1. LBBB will result in the left ventricle depolarizing after the right ventricle. Associated features of LBBB include long R waves in leads V5 and V6 and a long S wave in lead V1 (Hamad, 2018).

STD and STE are the most widely used features for detection of ischemic disease and myocardial infarction (MI), which is measured as the height difference between the J point and the reference line. The J point is at the end of the QRS complex and the beginning of the ST segment. The PR segment is used as the reference line for measuring the deviation of the ST segment. It is STE if the J point is 0.2 mV higher than the baseline, and STD if the J point is 0.05 mV lower than the baseline in leads V2 and V3 (O'Gara et al., 2013; Hanna and Glancy, 2015; Gupta et al., 2020). V5 is selected because it has the highest sensitivity in detecting myocardial ischemia (Crawford et al., 1999). Lead aVL is more reasonable for diagnosing MI caused by left anterior descending (LAD) coronary artery occlusion, especially extensive anterior MI (Acharya et al., 2019).

The characteristic of AF is small waves of high frequency (350–600 bpm). The diagnosis of AF is the absence of P waves in all leads and short, irregular RR intervals. Atrial flutter and AF are related arrhythmias and often have similar appearance. The distinct features of AF are the totally irregular rhythm and variable wave morphology, which are constant and identical, respectively, in atrial flutter (Goldberger et al., 2017).

AVB is characteristic of the prolonged PR interval. I-AVB occurs when the PR interval is ≥0.20 s. The associated clinical diagnosis criteria also include the electrical axis of the QRS complex. The normal mean QRS axis in adults lies in [−30°, +100°], and the left deviation of the electric axis (<−30°) is a noteworthy manifestation (Goldberger et al., 2017).

PAC can be diagnosed based on the P wave characteristics. Compared with the sinus P wave, a premature P wave has a different morphology and axis. A reverse P wave in lead II or III is a sign of PAC. In addition, it occurs earlier than the sinus P wave. A prolonged PR interval increases the probability of PAC. Lead aVR is used in detection (Gorgels et al., 2001).

PVC is recognized from a QRS complex that is wide (≥0.12 s) and abnormal in appearance. The premature ventricular impulse will replace a sinus beat and disrupt the regular interval between beats, which will lead to a prolonged RR interval.

The associated features are summarized in Table 2.

4.3 Dataset

In this work, the dataset used is obtained from the China Physiological Signal Challenge (CPSC) (Liu FF et al., 2018), which includes 9831 12-lead ECG recordings sampled at 500 Hz. The training set is open to the public and the testing set is private. To validate our model with more data and augment the dataset to reduce class imbalance, we incorporate the PTB-XL database (Wagner et al., 2020). The records are shown in Table 3.

Table 2 ECG features extracted based on domain knowledge

Feature	Description
$\mathrm{RR_{med}}$	Median of the RR interval
$\mathrm{RR_{min}}$	Minimum of the RR interval
$\Delta\mathrm{RR_{min}}$	Minimum of the difference of successive RR intervals
$\mathrm{RR_{std}}$	Standard deviation of the RR interval
pNN50	Percentage of NN interval differences ≥50 ms
$\mathrm{NN_{avg}}$	Average of NN intervals
$\mathrm{HR_{max}}$	Maximum of the heart rate
$\mathrm{HR_{min}}$	Minimum of the heart rate
R_{ApEn}	Approximate entropy of R peak amplitude
R_{med}	Median of R peak amplitude
R_{std}	Standard deviation of R peak amplitude
T_{PE}	Permutation entropy of T peak amplitude
T_{med}	Median of T peak amplitude
P_{PE}	Permutation entropy of P peak amplitude
P_{ApEn}	Approximate entropy of P peak amplitude
P_{med}	Median of P peak amplitude
P_{std}	Standard deviation of P peak amplitude
$\mathrm{ST_{dev}}$	Average deviation of the ST segment
$\mathrm{ST_{max}}$	Maximum deviation of the ST segment
$\mathrm{ST_{inter}}$	Deviation of slope intercept of the ST segment

Table 3 Number of recordings of datasets

Arrhythmia	Number of recordings	
	CSPC	PTB-XL
NSR	918	–
AF	1098	1514
I-AVB	704	797
LBBB	207	536
RBBB	1695	542
PAC	556	398
PVC	672	–
STD	825	–
STE	202	–

To reduce the effect of class imbalance, we randomly divide the records of each class into five subsets and copy the records of the class with fewer records so that the number of records of each class is nearly equal. The five subsets are processed to perform five cross validations.

We divide the public accessible records at a ratio of 70%:10%:20% randomly for training, validation, and testing, respectively. Every recording is labeled as the normal type or one of the eight abnormal types. For a recording with more than one label, the classification result is considered correct if it is consistent with one of the labels. Before being fed into the model, all the ECG signals are denoised and filtered to remove baseline wander using a Daubechies 6 wavelet (Singh and Tiwari, 2006).

The DNN model requires the input signal be a fixed segment. The length of CPSC recording varies from 6 to 60 s. The standard 12-lead ECG recording length is 10 s. These raw signals are preprocessed to a fixed length of 10 s. For shorter recordings, we pad shorter recording to achieve 10 s with data points copied from the same recording; for longer recordings, we split the long signal into several segments with a length of 10 s and input only one segment into the model. To prevent the model from overfitting, we input the different segments of the same recording in a different training epoch. There are 5000 preprocessed signal samples for each channel.

Signal cropping will inevitably lead to loss of information. That is why we use the complete record for the knowledge inference module. The records do not need cropping, but do need further slicing into heartbeats to extract domain features. The ECG signals are segmented according to the location of the R peak using the Pan–Tompkins algorithm (Pan and Tompkins, 1985), which is regarded as the identification of a heartbeat. The length of each heartbeat is fixed at 600 ms (200 ms before the R peak and 400 ms after) with 300 sample points. The features described in Table 2 are computed based on the heartbeat segmentation.

5 Results and discussions

The proposed model is developed and trained using Python with the TensorFlow library (Abadi et al., 2016). The experiments are performed on a computer with one Intel Core i9-9900K CPU at 3.6 GHz, NVIDIA Quadro RTX5000, and 64 GB memory. The Adam optimization method (Kingma and Ba, 2015) is used to optimize the model with the learning rate=0.001, beta1=0.9, and beta2=0.999. The procedure is conducted five times to complete the fivefold training and validation plus test.

5.1 Classification performance

In our experiments, the performance of the proposed model is evaluated with the following statistical measures as shown in Eqs. (10)–(13): sensitivity (Sen), specificity (Spe), precision (Pre), and accuracy (Acc). Sen measures the ability of the model to avoid missing an abnormal heartbeat, and Spe evaluates how well our model avoids misjudging a normal heartbeat. Pre measures the correctly predicted positive observations. Acc represents the overall performance of the model in properly classifying a heartbeat. True positive (TP) and true negative (TN) indicate the numbers of heartbeats correctly predicted, while false positive (FP) and false negative (FN) indicate the numbers of heartbeats not predicted as labeled.

$$\text{Sen} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{10}$$

$$\text{Spe} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \tag{11}$$

$$\text{Pre} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{12}$$

$$\text{Acc} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}. \tag{13}$$

For each class x, the F_1 score is denoted as F_{1x} and computed using Eq. (14), and the average F_1 score of the model is evaluated as Eq. (15):

$$F_{1x} = \frac{2(\text{Sen}\cdot\text{Pre})}{\text{Sen} + \text{Pre}}, \tag{14}$$

$$F_1 = \frac{1}{9}\sum_{x=1}^{9} F_{1x}. \tag{15}$$

The performance is shown in Table 4.

Table 4 Performance of the proposed model

Class	Sen	Spe	Pre	Acc	F_1
NSR	0.873	0.983	0.894	0.968	0.883
AF	0.928	0.989	0.946	0.979	0.936
I-AVB	0.850	0.997	0.970	0.981	0.906
LBBB	0.911	0.995	0.863	0.992	0.886
RBBB	0.985	0.983	0.954	0.983	0.899
PAC	0.900	0.990	0.900	0.982	0.900
PVC	0.950	0.990	0.914	0.986	0.902
STD	0.915	0.988	0.920	0.979	0.891
STE	0.826	0.993	0.789	0.988	0.887

To evaluate the effectiveness of our proposed model structure, we compare the performance measures of the proposed model with those of two other models. The first model (denoted as Expert in Table 5) uses the domain features described in Table 2 as the input of a classifier. We build a logistic regression on the extracted features. The second model (denoted as DNN) uses the DNN model described in Fig. 2, which uses convolutional neural network (CNN) blocks to extract the features of each lead, concatenates all 12 feature vectors together with a fully connected layer, then inputs the concatenated feature vectors to the classification layer, and outputs the probability distribution of the arrhythmia type. The F_1 scores of the three models are shown in Table 5.

Table 5 F_1 score in form "Mean±STD" of different models in the fivefold cross-validation

Class	F_1 score		
	Expert	DNN	Proposed
NSR	0.735±0.009	0.873±0.031	0.883±0.017
AF	0.802±0.016	0.821±0.027	0.916±0.014
I-AVB	0.809±0.014	0.828±0.017	0.901±0.011
LBBB	0.807±0.020	0.861±0.012	0.886±0.012
RBBB	0.701±0.015	0.821±0.024	0.894±0.009
PAC	0.725±0.043	0.860±0.028	0.901±0.011
PVC	0.849±0.023	0.884±0.012	0.902±0.007
STD	0.680±0.045	0.812±0.021	0.918±0.011
STE	0.797±0.021	0.842±0.030	0.889±0.007
Average	0.767±0.029	0.845±0.014	0.851±0.009

5.2 Effect of domain knowledge on performance

To demonstrate the effect of domain knowledge on the performance of the classifier more directly, the confusion matrices without and with domain knowledge are shown in Tables 6 and 7, respectively. The confusion matrix records the actual and predicted classifications for each class and identifies the type of errors being made by the classifier. The row labels indicate the true class records to which each row belongs, and the column labels indicate the class predicted by our model for records in each column. Numbers in each grid show the number of records classified as the column label when its true class is indicated by the row label.

In the classification of ECG arrhythmia, there are some domain-specific issues making the result unsatisfactory, leaving space to introduce the augmentation

Table 6 The confusion matrix of the DNN model

Class	Predicted label								
	NSR	AF	I-AVB	LBBB	RBBB	PAC	PVC	STD	STE
NSR	344	1	3	1	2	6	2	20	15
AF	2	428	4	2	13	10	4	5	1
I-AVB	11	8	256	4	10	7	3	1	1
LBBB	1	1	0	82	2	1	1	1	1
RBBB	3	1	0	2	718	2	2	1	0
PAC	5	8	1	0	4	225	6	1	0
PVC	0	2	1	0	4	6	267	1	0
STD	11	3	2	4	0	3	6	324	1
STE	8	1	0	0	3	0	1	2	71

Table 7 The confusion matrix of the proposed model

Class	Predicted label								
	NSR	AF	I-AVB	LBBB	RBBB	PAC	PVC	STD	STE
NSR	357	1	2	1	2	5	2	14	10
AF	1	440	2	3	8	7	4	2	1
I-AVB	9	8	263	4	6	6	3	1	1
LBBB	1	0	0	83	2	1	1	1	1
RBBB	2	1	0	2	720	1	2	1	0
PAC	3	3	1	0	2	238	1	1	1
PVC	0	1	1	0	1	2	275	1	0
STD	3	0	2	2	0	2	5	339	0
STE	1	0	0	0	2	0	1	2	79

of domain knowledge. The issues can be summarized as follows:

1. The influence of lost input data information: DNN models require input data be preprocessed into segments of a fixed length, which may lead to loss of important information. For PAC or PVC, the premature beat appears just a few times in the record, while other arrhythmias, such as AF, appear in each ECG beat. In extreme cases, AF beat appears only once. For the DNN model, the beat will be neglected because the record may be cropped and the characteristic beats are abandoned. In this case, the record will be misclassified in the NSR. We remedy this issue with the knowledge module, which takes the complete record as the input without cropping. The module magnifies the importance of specific important concepts missing from the learning model.

2. The influence of the similarity among classes: The similarity among classes will lead to high false positive cases. From the confusion matrix of the DNN model in Table 6, we can see that the DNN model is not sensitive to STE and STD detection. The small change of the ST segment amplitude is easily affected by noise, baseline drift, and subject variability. STD and STE can be misclassified into NSR, which makes their recognition from the training set a difficult task. The characteristic rules of specific leads aim to reduce the misclassification. Similarly, for the further classification of AF and atrial flutter, which are often misclassified for the morphology similarity, the difference between heart rates can be used as a distinguishing rule.

3. The influence of features of different importance: One important DNN model issue is that the influence of one feature is trivial and may be neglected if other features are normal. For example, atrial rhythm and sinus rhythm are easily confused. The pathological characteristic is P-wave anomaly. It is hard to distinguish when the amplitude of the P wave of a specific subject is very small and other features fall into the normal range. However, the logic rules can amplify the significance of a specific feature, and thus focus on the most discriminative part of the signal.

5.3 Trade-off between the DNN module and knowledge module

The hyperparameter λ in Eq. (1) creates a trade-off between the impact of the DNN module and knowledge module. It is sampled from a Beta distribution (Beta(β, β)). The hyperparameter is selected by observing the best F_1 performance on the validation set, as shown in Fig. 5. We test the model performance under different choices with β=0.1 and set λ to 0.1.

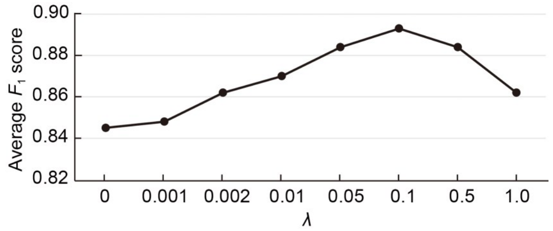

Fig. 5 Hyperparameter search for λ

When λ=0, the model regresses to a traditional CNN model. As λ grows, the performance is improved, which shows that logical rules of the knowledge module are essential for fallible categories with very similar patterns or ignored features. However, a too large λ will lead to reduced performance, because the power of automatically extracting nonlinear relation of the neural network may be significantly weakened by the logical rules, leading to high sensitivity and low precision. In addition, the knowledge module is domain-specific and is highly constrained by classification accuracy and representation power, and thus the parameter will impact the generalizability of the model.

In summary, a proper weight of the domain knowledge module is helpful in unifying the advantages of neural networks and logic reasoning. It should be estimated in a task-specific way.

5.4 Model parameter optimization

The learning rate and batch size impact the performance of the model. We conduct two contrast experiments: one experiment involves a different learning rate and an unchanged batch size, and the other involves a changed batch size with a fixed learning rate of 0.001.

The model is trained for a total of 50 epochs. Fig. 6 presents the loss curves with the batch size of 64. We test the learning rate of 0.01, 0.001, and 0.0001, and find that the model converges to a very low value with an increased epoch number and a different learning rate. With a learning rate of 0.001, the loss curve shows a stable convergence trend close to the value of 0, while the two other curves exhibit fluctuations during training.

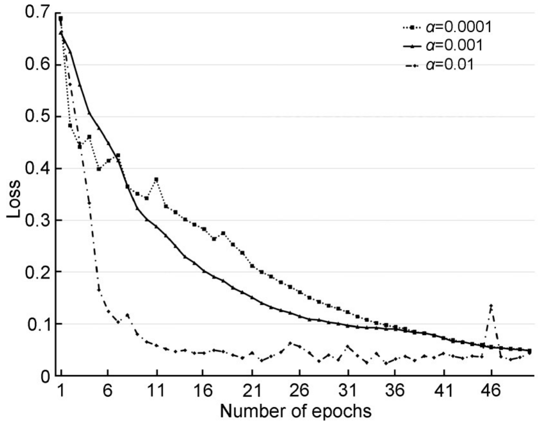

Fig. 6 The loss curves at different learning rates

By fixing the learning rate at 0.001, we test the model with different batch sizes. As illustrated in Table 8, the best performance is achieved at the batch size of 64. When the batch size is larger than 64, the F_1 score decreases as the batch size increases.

Table 8 Performance when using different batch sizes

Batch size	16	32	64	128
F_1	0.845	0.881	0.893	0.873

The average running time is about 70 s. Note that the model converges in a few minutes, also depending on the size and structure of the knowledge inference rules. The inference rules are designed in a concise and clear way to avoid recursive inference. Fortunately, rules in ECG classification are different from commonsense reasoning. For example, given two facts “Tom is Alice’s wife” and “John is Tom’s son,” a new fact, “John is Alice’s son,” can be deduced and the process can keep working until no new fact is generated. This technique is called forward chaining, and will result in a deep proof path. The training time will depend on the scale of the proof path. ECG classification rules avoid the issue because two arrhythmias or more will not infer the presence of a new arrhythmia.

5.5 Comparison with state-of-the-art methods

We conduct a comparative study of the proposed method and the state-of-the-art methods. The most frequently used neural networks in ECG classification tasks include CNNs, recurrent neural networks (RNNs), and their combination, convolutional recurrent neural networks (CRNNs).

CNNs have proved to be a very powerful and effective model in extracting sophisticated features, and are popular in different classification tasks including ECG signal classification. The ECG signal is sampled to be time series, so one-dimensional convolutional neural network (1D-CNN) is a preferred option. Although ECG segments can be transformed into two-dimensional representation to adapt to the conventional network, we still take time series as input to avoid introducing confounding factors and facilitate performance comparison. We conduct experiments with three popular CNN models as listed in Table 9: InceptionTime (Fawaz et al., 2020) (INCE for short), ResNet (He et al., 2016), and VGGNet (Simonyan and Zisserman, 2015). The model inputs are tensor of 5000×12 and the last FC layer is re-adapted to exclusively work with nine classes. ResNet includes one convolutional layer, eight residual blocks with two convolutional layers per block, and one FC layer. A kernel size of 5 is used in the 1D convolutions. VGGNet includes 16 1D convolutional layers with a kernel size of 3. INCE includes six inception blocks with kernel sizes of 40, 20, and 10 in each block. The experiment details are the same as in our experiment setup.

RNNs are natural for time-series data. We investigate long short-term memory (LSTM) (Mostayed et al., 2018), which comprises two hidden recurrent layers with 100 recurrent cells each and one FC classification layer. In most cases, an RNN is applied in combination with a CNN, i.e., CRNN, where the CNN is used as the feature extractor and the RNN is used to catch the time dependence of the time series. There are three different structures: CNN with LSTM (Luo et al., 2019), CNN with GRU (Chen TM et al., 2020), and CRNN with the attention module (Yao et al., 2020). To make the comparative study valid and sound, we select the studies using the same dataset and with approximately equal network depths.

Table 9 shows recent ECG classification results with bold data denoting the best performance. The experiment results show that, although the proposed model is not the best for some specific classes, it achieves the highest average F_1 score. The arrhythmia classes with the greatest performance improvement are PAC, PVC, STD, and STE. STD and STE could be misclassified as NSR without focusing on the deviation of the ST segment. PVC and PAC are characteristic of the premature beat, which occurs arbitrarily in an ECG recording. A fixed-length input of the CNN may lead to characteristic information loss and make it similar to the normal class. The knowledge module compensates for this by taking advantage of a domain-specific determinant.

The next two best models are ResNet and CRNN with an attention mechanism. In comparison with the two models, our work achieves an increase of 5.4% and 9.8% on average, respectively.

Table 9 Performance comparison between the proposed method and the state-of-the-art methods

Class	Average F_1 score							
	CNN			RNN	CRNN			Proposed
	INCE	ResNet	VGGNet	LSTM	CNN+LSTM	CNN+GRU	CNN+RNN+Attention	
NSR	0.717	**0.893**	0.783	0.738	0.806	0.795	0.790	0.883
AF	0.889	0.900	0.890	0.768	0.918	0.897	0.930	**0.936**
I-AVB	0.872	0.850	0.841	0.741	0.881	0.865	0.850	**0.901**
LBBB	0.841	0.874	0.872	0.705	**0.900**	0.821	0.860	0.886
RBBB	0.854	0.922	0.901	0.821	0.925	0.911	**0.930**	0.894
PAC	0.798	0.849	0.703	0.590	0.845	0.734	0.750	**0.901**
PVC	0.786	0.776	0.738	0.807	0.727	0.852	0.850	**0.902**
STD	0.783	0.762	0.721	0.658	0.782	0.788	0.800	**0.918**
STE	0.704	0.797	0.549	0.294	0.615	0.509	0.560	**0.889**
Average	0.805	0.847	0.771	0.680	0.809	0.797	0.813	**0.893**

Best performances are in bold

The LSTM model alone does not perform well on the task, but the combination with a CNN leads to significant performance improvement due to the excellent power of extracting nonlinear CNN features. Note that we do not examine the RNN model with carefully designed input, which might achieve competitive performance as their convolutional counterparts.

In summary, our model attains similar or competitive results when compared to the available state-of-the-art models. Learning with knowledge injection will produce more representative features, thus avoiding overfitting. The rich feature space in the process of knowledge injection learning improves the sensitivity and specificity of the model. Compared with the above-mentioned methods, we believe that infusion of domain knowledge into the DNN model will reduce false alarms, improve interpretability, and provide robustness for practical applications.

6 Conclusions

In this study, we propose an automatic classification model for cardiac arrhythmia that combines DNN and domain knowledge. The model consists of a DNN to capture the statistical pattern between input data and the ground-truth label, and a knowledge module to guarantee consistency with the domain knowledge. These two components are trained interactively to bring the best of both worlds.

Our method answers the questions raised in Section 1 as follows: (1) Domain knowledge is represented by fuzzy logic rules, which can map a proposition into a real value in the range [0,1], making the truth degree comparable to the probability vector. (2) Logic rules are indifferentiable but can be relaxed using the t-norm, so the derivation can be computed and the gradient descent method can be applied to train the model jointly. (3) The performance is improved because the knowledge inference module reduces the influence of lost input data information, similarity between classes, and features of different importance. Compared to the end-to-end DNN model, the F_1 score of each arrhythmia of the knowledge-enhanced model increases, which means that the domain knowledge is helpful in learning information that the neural network cannot exploit.

We have instantiated our method for the ECG arrhythmia classification task. The experiment shows that our model attains competitive results when compared to many existing approaches. The method can be applied to other decision-making fields to provide generalization, reduce data bias, and improve interpretability.

Compliance with ethics guidelines

Jie SUN declares that he has no conflict of interest.

Data availability

The data that support the findings of this study are openly available in China Physiological Signal Challenge 2018 at http://2018.icbeb.org/Challenge.html and PTB-XL database at https://physionet.org/content/ptb-xl/1.0.1/.

References

Abadi M, Barham P, Chen JM, et al., 2016. TensorFlow: a system for large-scale machine learning. 12th SENIX Conf on Operating Systems Design and Implementation, p.265-283.

Acharya UR, Fujita H, Oh SL, et al., 2019. Deep convolutional neural network for the automated diagnosis of congestive heart failure using ECG signals. *Appl Intell*, 49(1): 16-27. https://doi.org/10.1007/s10489-018-1179-1

Baloglu UB, Talo M, Yildirim Ö, et al., 2019. Classification of myocardial infarction with multi-lead ECG signals and deep CNN. *Patt Recogn Lett*, 122:23-30. https://doi.org/10.1016/j.patrec.2019.02.016

Chen B, Guo W, Li B, et al., 2018. A study of deep feature fusion based methods for classifying multi-lead ECG. https://arxiv.org/abs/1808.01721

Chen TM, Huang CH, Shih ESC, et al., 2020. Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model. *iScience*, 23(3): 100886. https://doi.org/10.1016/j.isci.2020.100886

Crawford MH, Bernstein SJ, Deedwania PĆ, et al., 1999. ACC/AHA guidelines for ambulatory electrocardiography: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Committee to revise the guidelines for ambulatory electrocardiography) developed in collaboration with the North American Society for Pacing and Electrophysiology. *J Am Coll Cardiol*, 34(3):912-948. https://doi.org/10.1016/S0735-1097(99)00354-X

Dai WZ, Xu QL, Yu Y, et al., 2019. Bridging machine learning and logical reasoning by abductive learning. 33rd Conf on Neural Information Processing Systems, p.2811-2822.

Diligenti M, Gori M, Sacca C, 2017. Semantic-based regularization for learning and inference. *Artif Intell*, 244: 143-165. https://doi.org/10.1016/j.artint.2015.08.011

Fawaz HI, Lucas B, Forestier G, et al., 2020. InceptionTime:

finding AlexNet for time series classification. *Data Min Knowl Disc*, 34(6):1936-1962. https://doi.org/10.1007/s10618-020-00710-y

Giannini F, Diligenti M, Gori M, et al., 2019. On a convex logic fragment for learning and reasoning. *IEEE Trans Fuzzy Syst*, 27(7):1407-1416. https://doi.org/10.1109/TFUZZ.2018.2879627

Goldberger AL, Goldberger ZD, Shvilkin A, 2017. Goldberger's Clinical Electrocardiography (9th Ed.). Elsevier, Armstrong, the Netherlands.

Gorgels APM, Engelen DJM, Wellens HJJ, 2001. Lead aVR, a mostly ignored but very valuable lead in clinical electrocardiography. *J Am Coll Cardiol*, 38(5):1355-1356. https://doi.org/10.1016/S0735-1097(01)01564-9

Gupta A, Huerta EA, Zhao ZZ, et al., 2020. Deep learning for cardiologist-level myocardial infarction detection in electrocardiograms. Proc 8th European Medical and Biological Engineering Conf, p.341-355. https://doi.org/10.1007/978-3-030-64610-3_40

Hamad T, 2018. ABC of Clinical Electrocardiography (2nd Ed.). BMJ Books, Massachusetts, USA.

Hanna EB, Glancy DL, 2015. ST-segment elevation: differential diagnosis, caveats. *Cleveland Clin J Med*, 82(6):373-384. https://doi.org/10.3949/ccjm.82a.14026

Hannun AY, Rajpurkar P, Haghpanahi M, et al., 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat Med*, 25(1):65-69. https://doi.org/10.1038/s41591-018-0268-3

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/CVPR.2016.90

Hu ZT, Ma XZ, Liu ZZ, et al., 2016. Harnessing deep neural networks with logic rules. Proc 54th Annual Meeting of the Association for Computational Linguistics, p.2410-2420. https://doi.org/10.18653/v1/P16-1228

Ioffe S, Szegedy C, 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proc 32nd Int Conf on Machine Learning, p.448-456.

Jin LP, Dong J, 2017. Classification of normal and abnormal ECG records using lead convolutional neural network and rule inference. *Sci China Inform Sci*, 60(7):078103. https://doi.org/10.1007/s11432-016-9047-6

Kimmig A, Bach SH, Broecheler M, et al., 2012. A short introduction to probabilistic soft logic. Proc 11th NIPS Workshop on Probabilistic Programming: Foundations and Applications, p.1-4.

Kingma DP, Ba LJ, 2015. Adam: a method for stochastic optimization. Proc 3rd Int Conf on Learning Representations. https://doi.org/10.48550/arXiv.1412.6980

Klir GJ, Yuan B, 1995. Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall, New Jersey, USA.

Liu FF, Liu CY, Zhao LN, et al., 2018. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *J Med Imag Health Inform*, 8(7):1368-1373. https://doi.org/10.1166/jmihi.2018.2442

Liu WH, Zhang MX, Zhang YD, et al., 2018. Real-time multilead convolutional neural network for myocardial infarction detection. *IEEE J Biomed Health Inform*, 22(5):1434-1444. https://doi.org/10.1109/JBHI.2017.2771768

Luo CS, Jiang HX, Li QC, et al., 2019. Multi-label classification of abnormalities in 12-lead ECG using 1D CNN and LSTM. Proc 8th Int Workshop on Machine Learning and Medical Engineering for Cardiovascular Healthcare, p.55-63. https://doi.org/10.1007/978-3-030-33327-0_7

Mostayed A, Luo JY, Shu XL, et al., 2018. Classification of 12-lead ECG signals with bi-directional LSTM network. https://arxiv.org/abs/1811.02090

National Center for Cardiovascular Diseases, 2019. Report on Cardiovascular Diseases in China 2018. Encyclopedia of China Publishing House, Beijing, China (in Chinese).

O'Gara PT, Kushner FG, Ascheim DD, et al., 2013. ACCF/AHA guideline for the management of ST-elevation myocardial infarction: a report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines. *Circulation*, 127(4):e362-e425. https://doi.org/10.1161/CIR.0b013e3182742cf6

Pan JP, Tompkins WJ, 1985. A real-time QRS detection algorithm. *IEEE Trans Biomed Eng*, BME-32(3):230-236. https://doi.org/10.1109/TBME.1985.325532

Parvaneh S, Rubin J, Rahman A, et al., 2018. Analyzing single-lead short ECG recordings using dense convolutional neural networks and feature-based post-processing to detect atrial fibrillation. *Physiol Meas*, 39(8):084003. https://doi.org/10.1088/1361-6579/aad5bd

Sankaran PG, Sunoj SM, Nair NU, 2016. Kullback–Leibler divergence: a quantile approach. *Stat Probab Lett*, 111:72-79. https://doi.org/10.1016/J.SPL.2016.01.007

Simonyan K, Zisserman A, 2015. Very deep convolutional networks for large-scale image recognition. Proc 3rd Int Conf on Learning Representations. https://doi.org/10.48550/arXiv.1409.1556

Singh BN, Tiwari AK, 2006. Optimal selection of wavelet basis function applied to ECG signal denoising. *Dig Signal Process*, 16(3):275-287. https://doi.org/10.1016/j.dsp.2005.12.003

Singstad BJ, Tronstad C, 2020. Convolutional neural network and rule-based algorithms for classifying 12-lead ECGs. Computing in Cardiology, p.1-4. https://doi.org/10.22489/CinC.2020.227

Surawicz B, Knilans TK, 2008. Chou's Electrocardiography in Clinical Practice: Adult and Pediatric (6th Ed.). Saunders Elsevier, Philadelphia, USA.

Wagner P, Strodthoff N, Bousseljot RD, et al., 2020. PTB-XL, a large publicly available electrocardiography dataset. *Sci Data*, 7(1):154. https://doi.org/10.1038/s41597-020-0495-6

Yao QH, Wang RX, Fan XM, et al., 2020. Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network. *Inform Fus*, 53:174-182. https://doi.org/10.1016/J.INFFUS.2019.06.024

Zhou FY, Jin LP, Dong J, 2017. Premature ventricular contraction detection combining deep neural networks and rules inference. *Artif Intell Med*, 79:42-51. https://doi.org/10.1016/j.artmed.2017.06.004