Information Sciences 581 (2021) 42–59

Contents lists available at ScienceDirect

Information Sciences

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

Interactive ECG annotation: An artificial intelligence method for smart ECG manipulation

Haiyan Wang [a,b,c], Yanjie Zhou [d,*], Bing Zhou [c,e], Xiangdong Niu [f], Hua Zhang [g], Zongmin Wang [a,c]

[a] *State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450003, China*
[b] *Simulation Experiment Centre, Zhengzhou University of Aeronautics, Zhengzhou 450046, China*
[c] *Collaborative Innovation Centre for Internet Healthcare, Zhengzhou University, Zhengzhou 450052, China*
[d] *School of Management Engineering, Zhengzhou University, Zhengzhou 450001, China*
[e] *School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China*
[f] *The Fifth Affiliated Hospital of Zhengzhou University, Zhengzhou University, Zhengzhou 450000, China*
[g] *Department of Cardiology, Zhengzhou Seventh People's Hospital, Zhengzhou 450000, China*

ARTICLE INFO

Article history:
Received 9 February 2021
Received in revised form 26 August 2021
Accepted 29 August 2021
Available online 2 September 2021

Keywords:
Intelligent ECG annotation
Intelligent simulation data
GAN
Beat pre-annotation
CNN

ABSTRACT

An electrocardiogram (ECG) consists of complex segments, such as P-QRS-T waves. Manual ECG annotation is challenging and time-consuming, even for specialist physicians. The shortage of labelled ECG data is one of the essential factors that affect ECG intelligent analysis's long-term development. This study proposes an intelligent ECG-assisted annotation system, that not only supplements labelled data, but also significantly reduces the workload compared with manual annotation. Since beat annotation is the most basic and important part, a GAN-based generation model that can generate 14 types of simulation beats and a CNN-based beat pre-annotation model are proposed. The experimental results show that the simulation beat has high similarity to real beat and the accuracy of the pre-annotation model on the test set of 14 classes of beats is 99.28%. The proposed ECG intelligent annotation system's self-learning mechanism could improve pre-annotation performance and annotation efficiency by generating more labelled data. The proposed annotation system can also be extended to other data annotation applications.

1. Introduction

With the development of information technology, computer-assisted medical treatment has attracted increasing attention from academia and industry to free doctors from tedious work. Increased numbers of cardiovascular patients face the problem of a lack of specialist physicians to diagnose ECG in most hospitals, especially in China. The ECG-assisted diagnosis system requires accurate and real-time feedback, which has high performance and operating efficiency requirements. Computer hardware systems are sufficient to meet the auxiliary diagnosis model's calculation and storage requirements with computer science development. However, the lack of labelled ECG data has slowed down the building of excellent auxiliary diagnosis models. Although ECG diagnosis generates a large amount of ECG data every day, they are all unlabelled and

[*] Corresponding author.
E-mail address: ieyjzhou@zzu.edu.cn (Y. Zhou).

https://doi.org/10.1016/j.ins.2021.08.095
0020-0255/© 2021 Elsevier Inc. All rights reserved.

cannot be used for model building directly. The labelled data are insufficient in terms of quantity and types, which hinders the development of the computer-assisted diagnosis field to a certain extent.

In ECG intelligent analysis, many factors lead to the shortage of labelled ECG data. First, ECG data are sensitive and difficult to collect because of medical data confidentiality. Second, building a standard ECG database is a foundation work that requires long-term human and financial investment. Finally, the critical factor is the difficulty of ECG annotation. The ECG annotation is cumbersome, requiring a professional annotation system and manual annotation by specialist physicians from beat to beat. ECG annotation includes beat, rhythm, morphology, and conclusive annotation. Fig. 1 shows a typical annotation example, which provides the beat and rhythm labels. The annotation example in Fig. 2 contains beat, rhythm, and morphological labels. Even a short-term ECG, not only has a wide variety of labels but also requires accurate positioning and annotation.

Two ECG annotation software can be found. One is WAVE [1], an auxiliary annotation software developed by the Massachusetts Institute of Technology. The most widely used MIT-BIH Arrhythmia Database [2,3] is derived from WAVE. WAVE includes beat annotation, rhythm annotation, partial wavelet positioning, and key wavelet shape change prompts. WAVE is also a semiautomatic beat annotation system. First, all beats were automatically labelled as normal and then modified. The other auxiliary annotation software was developed based on the Chinese Cardiovascular Disease Database (CCDD) [4]. The annotation functions of this tool are the beat, morphology, and conclusive annotation. Both annotation systems have some limitations. First, they are both manually annotated, which is cumbersome and work-intensive. Second, they lack perfect management, which is not convenient for large-scale annotation. Finally, the annotation function is incomplete, for example, WAVE lacks complete morphological annotation, and CCDD auxiliary annotation software lacks rhythm annotation. Therefore, this paper proposes a human–machine integration ECG intelligent annotation system that integrates intelligent technology into the auxiliary annotation system. First, the ECG data are pre-annotated using artificial intelligence. Specialist physicians only need to review and revise the pre-annotation results, which can significantly reduce the workload and improve the efficiency.

The data size of prevalent diseases (abnormal), especially rare diseases, is often much less than the healthy data (normal). The imbalance of the data size between abnormal and normal will affect the classification performance of the model. Annotating more ECG data is the most straightforward solution, especially for rare disease data. However, cases of rare diseases are usually minimal. Therefore, we propose a solution that generates intelligent simulation data, which can be used as labelled data to assist in model building. It solves the problem of insufficient labelled data and guarantees the pre-annotation performance of minority classes. Generating sufficient and accurate data is conducive to training the model. The ECG generation methods can be divided into two categories. One category is the traditional method, which extracts signal features manually and constructs a generation model. For example, McSharry et al. [5] used coupled ordinary differential equations to generate a single beat signal dynamic model, Li et al. [6] used a data flow graph to describe the ECG signal, Sayadi et al. [7] designed a Gaussian wave-based state-space to model the temporal dynamics of ECG signals, and Roonizi et al. [8] introduced polynomial spline models for modeling. The abovementioned traditional methods need to extract features and adjust model parameters manually, and the quality of the generation model depends entirely on the human experience. The second category uses deep neural networks to generate ECG data. Golany et al. [9] extracted features and then input them to a GAN to synthesize ECG data. Zhu et al. [10] used a GAN to generate short-term ECG segments similar to clinical data directly, and the GAN is composed of a CNN and LSTM. When balancing the data set, the generated data must be accurate data corresponding to the unbalanced minority classes. Accuracy means that it can be used as a specific type of beat. Hernandez-Matamoros et al. [11] used the Bidirectional Recurrent Neural Network (BiRNN) model to synthesize multiple types of beat signals similar to the original data, but in the data preprocessing stage, strict signal segmentation and reconstruction are required. Wang et al. [12] designed a simple GAN that can generate accurate left bundle branch block beats without the need to reconstruct the original signal. However, this model is not universal for generating all types of beats and lacks reliable indicators to measure the generated data's quality. In this paper, we improved the model to apply to all

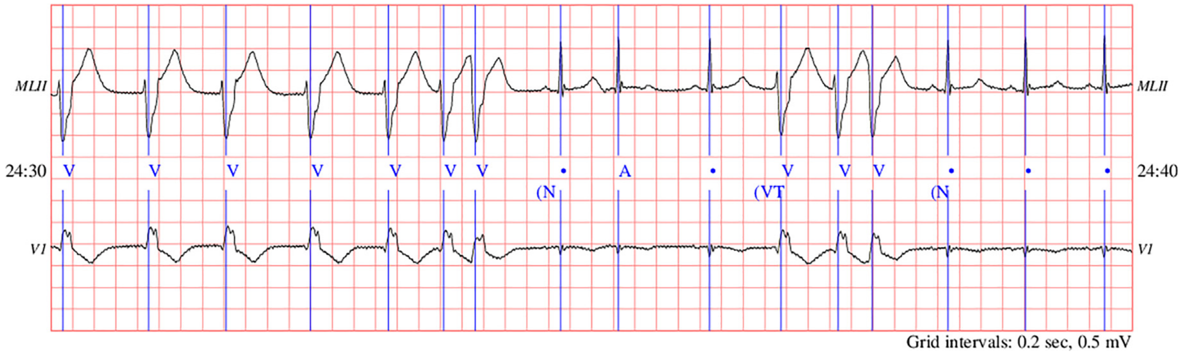
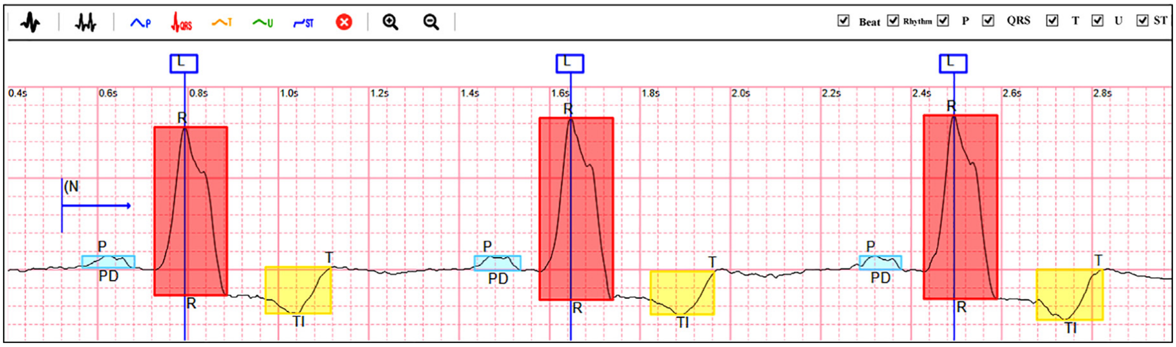


Fig. 1. A section of 10 s ECG data from record 205 of the MIT–BIH Arrhythmia Database. The figure contains two leads, the upper is the MLII lead, the lower is the V1 lead, and the middle is the labels, including the beat labels and the rhythm labels. Beat annotation interpretation: V → ventricular premature beat;·→normal beat; A → atrial premature beat. Rhythm annotation interpretation: (N → normal sinus rhythm; (VT → ventricular tachycardia. Source: Image taken from https://archive.physionet.org/cgi-bin/atm/ATM.



☑ Beat ☑ Rhythm ☑ P ☑ QRS ☑ T ☑ U ☑ ST

Fig. 2. The image is an annotation interface of the annotation system developed by this paper. The signal in the image is lead I, including beat, rhythm, and morphology labels. Beat annotation interpretation: L → Left bundle branch block beat. Rhythm annotation interpretation: (N → normal sinus rhythm. Morphological annotation interpretation: Above the waveform—P → P Wave; T → T Wave; R → R wave peak, representing QRS wave. Below the waveform—PD → P wave bimodal; R → The morphological type of this QRS complex is R; TI → T wave inversion.

types of beats and introduced the indicator maximum mean discrepancy (MMD) to evaluate the accuracy of the generated data.

The purpose of this paper is to develop an ECG intelligent annotation system based on generated simulation data, which can not only solve the problem of difficult annotation but also solve the problem of insufficient labelled data and imbalance of data sets during the model training. The main contributions of this paper are summarized as follows: (1) This paper proposes a human–machine integration ECG intelligent annotation system that can free annotation experts from heavy manual annotation. Annotation experts only need to review and verify the pre-annotation result. (2) This paper also introduces a GAN-based beat generation model that can generate 14 types of beats accurately. We measure the generation model's effectiveness and accuracy by the loss curve and quantitative indicator MMD and finally design a blind experiment to verify the availability of the generated data. (3) This paper proposes a CNN-based beat pre-annotation model based on generated data. The experimental results show that the performance of the proposed method is better than that of the state-of-art methods. The ECG intelligent annotation system proposed in this paper will accelerate the development of ECG computer-assisted diagnosis and can be extended to other intelligent medical fields.

This paper's organized structure is presented as follows: Section 2 introduces the ECG intelligent annotation system based on simulation data; Section 3 describes the beat pre-annotation algorithm. Section 4 discusses and compares the results of the proposed algorithms. Finally, conclusions and future research directions are presented in Section 5.

2. The proposed framework

This section introduces the proposed framework of human–machine integration ECG intelligent annotation, which is an efficient, intelligent system. The details of the proposed framework are presented as follows.

2.1. Framework overview

The ECG intelligent annotation system consists of three modules: the filtering module, the management module, and the auxiliary annotation module. The details of the modules are described as follows.

Filtering module: Filtering ECG samples that are worth annotating or intercepting a segment in an ECG record as the sample to be labelled according to data annotation requirements. Generally, the data selected have clinical research value, and the data are not polluted by serious noise.

Management module: Manage the entire annotation process, including annotation expert management, warehousing management of samples to be labelled, annotation task management (task packaging and task allocation), intelligent pre-annotation result review management, and labelled data storage management.

Auxiliary annotation module: The annotation includes beat annotation, rhythm annotation, morphological annotation, and conclusive annotation. The morphological annotation marked each wavelet's (P wave, QRS wave, T wave, and ST segment) shape, starting, and ending points.

The core idea of the proposed ECG annotation system is to integrate intelligent technology to solve the problem of onerous and low efficiency. Fig. 3 shows a flowchart drawn from the perspective of intelligent annotation, which describes the processes from the raw data to the labelled data into the ECG database. The detailed steps are selection and interception of samples to be annotated (put into the sample database to be annotated), annotation task packaging and distribution, sample pre-annotation, pre-annotation results review and modification, conclusive annotation, and labelled data put into the ECG database. On the other hand, labelled data will be fed back to the intelligent annotation system to train the intelligent models (the blue shading box in Fig. 3), significantly reducing the onerousness, and improving the efficiency and accuracy of anno-

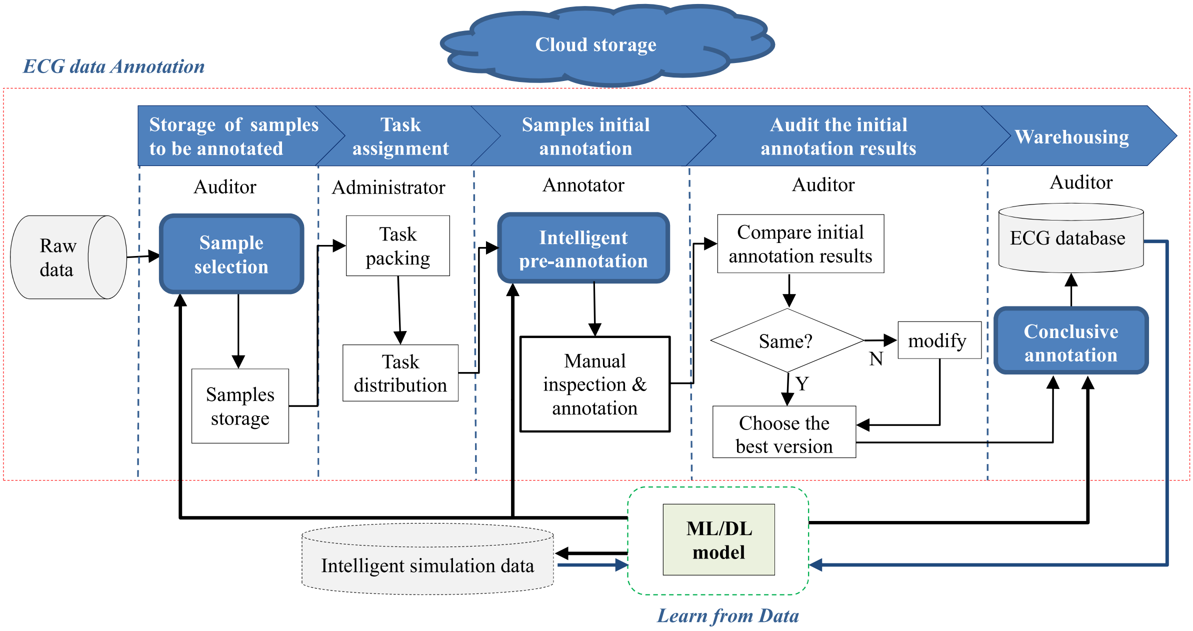


Fig. 3. The annotation process of the ECG intelligent annotation system based on simulation data generation. The "Intelligent pre-annotation" includes Beat pre-annotation, Rhythm pre-annotation, and Morphological pre-annotation.

tation. With the continuous increase in labelled data, the learning ability of the model becomes stronger. The intelligent applications in the system are summarized as follows:

a) **Generate intelligent simulation data:** Due to the imbalance of the incidence of various arrhythmia diseases, the training data set is unbalanced, which will greatly affect the pre-annotation performance. We introduce an artificial intelligence-based method to generate simulation data. First, we use the precious labelled ECG data to train the generation model and generate accurate ECG data. Then the generated data feedback to the system for balancing and supplementing the data set and assisting model training, such as sample filtering model and pre-annotation model. The ECG data generation is the basis of the intelligent annotation system.
b) **Intelligent selection and interception of samples to be annotated:** Selecting or intercepting valid ECG records or segments. The technology available for this task is intelligent detection of the ECG signal quality. ECG signals usually contain noise, such as baseline wanders, motion artifact, and muscle electricity [13]. Signals with severe pollution have no clinical diagnostic value and can be ignored directly. For signals with a large number of records or that are dozens of hours long, the workload is tremendous if the expert browses one by one. Thus, the signal quality can be graded using the ECG intelligent quality detection technology, and the screening experts only review the results.
c) **ECG intelligent pre-annotation**: Traditional annotation needs to manually label the type of each beat, identify the starting point and label the type of each rhythm, mark the start–end points and shape of each wavelet, which is onerous. Therefore, mature ECG intelligent diagnosis and analysis technology can be used for intelligent pre-annotation, such as the classification of beats, diagnosis of diseases [14], and positioning of waveforms. The annotation experts only need to review and modify the pre-annotation results.
d) **Conclusive annotation:** Summative diagnosis requires review experts to annotate all diseases in the sample, and multi-label classification technology based on arrhythmia diagnosis can be used. Experts only need to review and modify the conclusive pre-annotation results, which can improve the efficiency and reduce the rate of experts' missed annotations.

Although the system integrates intelligent technology to assist the annotation, it does not guarantee that the annotation results are error-free. Therefore, after intelligent screening and pre-annotation, expert review is required. The human–machine integration mode not only improves the efficiency but also guarantees the accuracy of the annotation. With increasingly more labelled data, the trained model's performance will improve, the efficiency will increase, and the system will be in a virtuous circle.

2.2. Highlights of the framework

The proposed system is intelligent and efficient and can meet various large-scale annotation requirements. The highlights of the proposed system are summarized as follows:

1) **Web-based management mode:** Compared to a stand-alone mode that does not require networking, the web-based management mode has a rigorous and smooth design, which can manage large-scale annotation effectively, including annotation task management, task assignment management, strict annotation process management, data management at all stages, and role management, etc. Annotation experts can download the annotation client compression package anytime and anywhere. To annotate data, the user only needs to unzip and log in, which maximizes make use of the expert's fragmented time.
2) **Complete annotation function:** This system has a complete annotation function, including beat, rhythm, morphology, and conclusive annotation. It can select annotation content flexibly and build ECG databases of different topics. It meets the requirements of traditional machine learning algorithms based on feature extraction and deep learning algorithms that only need conclusive annotation.
3) **Self-learning intelligent modules:** Self-learning means that the training and using of intelligent modules is a virtuous cycle learning process that inputs new knowledge continuously. As shown in Fig. 4, the labelled data annotated with the assistance of the intelligent module will be fed back to the system again for model adjustment and optimization. The optimized model can improve the pre-annotation performance and generate more accurate data. In this mode, the learning ability and performance of intelligent models are becoming increasingly stronger, the generated data are becoming increasingly accurate, the pre-annotation performance and annotation efficiency are increasing, and the workload of annotation experts is decreasing.
4) **Strict annotation process:** To ensure the accuracy of the annotation results, the system has designed a rigorous annotation process. First, each sample is annotated by two annotators (intermediate experts) simultaneously, and then the results are matched to show inconsistent content. Finally, senior experts select the best version to modify and input into the ECG database. This mode can guarantee the accuracy of the annotation results with a minimal workforce.
5) **Cloud storage mode:** Large-scale annotation requires high access speed and stability. The cloud storage mode can ensure the speed of data access and ensure the stability and security of the system.

3. Beat pre-annotation optimization

Beat annotation needs to identify the type of each beat, which is an essential part of ECG annotation. Beat annotation is more cumbersome than rhythm annotation and more necessary than morphological annotation, and thus, the pre-annotation of beat is especially important. The difference in disease incidence will cause extremely unbalanced data sets, affecting the classification results' accuracy significantly. Therefore, to solve unbalanced data sets in beat pre-annotation, we propose a beat pre-annotation method based on simulation data generation. This method uses simulation data generated by a GAN model to balance the data set and improve the beat prediction.

3.1. Preliminaries

3.1.1. Generative adversarial networks (GAN)

Generative adversarial networks, which was proposed by Google researcher Ian Goodfellow in 2014, belongs to unsupervised deep learning methods. The main structure of GAN includes a generator (G) and a discriminator (D), where G receives random noise z to generate data G(z) as real as possible, and D judges whether the data is real and tries to distinguish between real data × and fake data G(z). G and D confront each other and adjust the parameters constantly.

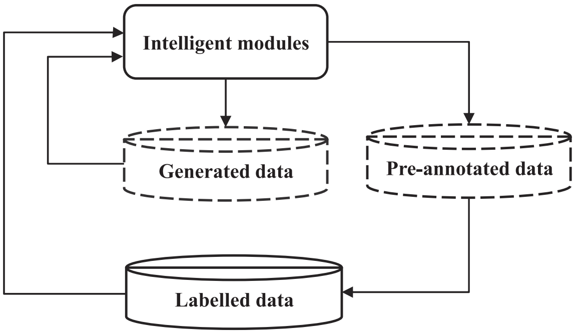


Fig.4. Self-learning mode of intelligent modules.

The optimization objective function V(D, G) of GAN is shown in equation (1):

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))], \tag{1}$$

where D(x) represents the probability that the real data × is real, and D(G(z)) represents the probability that the generated data G(z) is real. The probability range of D(G(z)) is [0,1], where 1 and 0 represent the real data and fake data, respectively. The best state of the game is D(G(z)) = 0.5, which means that the data generated by generator G cannot distinguish between true and false.

In this paper, the optimization functions of the discriminator D and the generator G can be obtained by formula (1). The optimization function of the discriminator D is shown in formula (2):

$$\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{2}$$

The optimization function of the generator G is shown in formula (3):

$$\min_{G} V(D,G) = E_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{3}$$

3.1.2. Convolutional neural networks (CNN)

A convolutional neural network is a feedforward neural network that has been widely used in deep learning. It can process one-dimensional and multidimensional data effectively, such as image data, time-series data, video data, and so on. The CNN is mainly composed of the input layer, convolutional layer, pooling layer, fully connected layer, and output layer [15]. The details of these layers are described as follows.

Convolutional layer: The function of the convolutional layer is to extract the characteristics of the input data. It is a local operation that contains multiple convolution kernels. Each element of the convolution kernel corresponds to a weight coefficient and a bias vector. The input feature area corresponding to the size of the convolution kernel is called the receptive field. The convolution kernel is generally square. There will be regular scanning of input features when working, and the input features will be multiplied by matrix elements and summed in the receptive field. Then the deviation is superimposed, as shown in formula (4).

$$Z^{l+1}(i,j) = \left[Z^l \otimes w^{l+1}\right](i,j) + b = \sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[Z_k^l(s_0 i + x, s_0 j + y)w_k^{l+1}(x,y)\right] + b$$

$$(i,j) \in \{0,1,\ldots L_{l+1}\} \quad L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \tag{4}$$

where b is the bias vector. Z^l and Z^{l+1} represent the (l + 1)th convolution layer's input and output, respectively. L_{l+1} is the size of Z^{l+1}, K is the number of channels of the feature map, while f, S_0, and p correspond to the size of the convolution kernel, the convolution stride, and the number of paddings, respectively.

Pooling: Pooling is a form of down-sampling. It has many different forms of nonlinear pooling functions. The essence is to replace the result of a single point with the feature map statistic of its neighboring regions, and "max pooling" is the most common. The pooling reduces the dimension under the condition that the feature is not deformed. It continuously reduces the data's space size, in such a way that the number of parameters and the amount of calculation will also decrease, which controls over-fitting to a certain extent. Pooling layers are periodically inserted into convolutional layers.

Fully connected layer: The fully connected layer in the CNN is equivalent to the traditional feedforward neural network's hidden layer. It is generally located in the last part of the network, and its role is to transmit signals to other fully connected layers.

The convolutional layer and pooling layer will generally have several, which are alternately arranged. A convolutional layer is connected to a pooling layer, and then a convolutional layer is connected after the pooling layer. Each neuron's input in the convolutional layer is connected with the previous layer's local area, and the local features are extracted. The pooling layer is the calculation layer used to find local sensitivity and secondary feature extraction. The two-time feature extraction structure reduces the feature resolution and reduces the number of parameters that must be optimized.

3.2. Data pre-processing

The most widely used MIT-BIH Arrhythmia Database [2,3] is selected as a standard data set to evaluate the proposed method. The database contains 48 ECG records with a duration of half an hour, collected from 47 subjects, and the sampling rate is 360HZ. Each ECG record consists of two leads. In this paper, one lead signal is selected for training and testing.

The ECG records of the database contain a wide variety of beats. To get a complete beat, 110 and 175 sampling points are taken from the left and right of the R wave peak (reference point). Hence a beat contains 286 sampling points. Finally, the extracted beats and their numbers are shown in Table 1. A total of 14 types of beats are extracted. The number of normal beats is 75019, while the supraventricular premature or ectopic beat (atrial or nodal) has only two, and the atrial escape beat is 16. The sample size of different beats varies greatly, which shows that the data set is extremely unbalanced.

Table 1
Beat distribution of MIT-BIH Arrhythmia Database.

Label ID	Original label	Label	Description	Number of real beats
0	N	N	Normal beat	75,019
1	L	L	Left bundle branch block beat	8072
2	R	R	Right bundle branch block beat	7255
3	e	e	Atrial escape beat	**16**
4	j	j	Nodal (junctional) escape beat	229
5	A	A	Atrial premature beat	2546
6	a	a	Aberrated atrial premature beat	150
7	J	J	Nodal (junctional) premature beat	**83**
8	S	S	Supraventricular premature or ectopic beat (atrial or nodal)	**2**
9	V	V	Premature ventricular contraction	7129
10	E	E	Ventricular escape beat	106
11	F	F	Fusion of ventricular and normal beat	802
12	/	p	Paced beat	7024
13	f	f	Fusion of paced and normal beat	982

Annotation: The "Label" column in the table1 is the annotation label of each type of beat defined in this study.

3.3. Beat pre-annotation method based on generated data

The beat intelligent pre-annotation method based on simulation generation data proposed in this paper is roughly divided into two parts: (1) generating accurate beats and balancing the data set and (2) training the pre-annotation model on the balanced data set and pre-annotated the beats. The process of this method is shown in Fig. 5:

Design a new GAN-based beat generation model: The detailed process is that generator G receives noise z to generate simulation data and passes it to discriminator D. D judges the authenticity of the generated data and feeds back the discrimination result to the network. The network is adjusted and optimized according to the feedback results until G can generate data that D cannot distinguish between real and fake, then, the GAN reaches the optimal state. Both generator G and discriminator D designed in the GAN are composed of a fully connected network. In the experiment, the input labelled data is the 14 types of labelled beats we extracted from the MIT-BIH Arrhythmia Database. The noise z input for generating data is a set of random numbers drawn from the standard normal distribution, and each generated beat is composed of 95 data points.

Design a new CNN-based beat pre-annotation model: The samples to be annotated are first pre-annotated by the beat intelligent pre-annotation model, then the pre-annotation result is reviewed and modified by the reviewing expert, and finally put into the ECG database. In the stage of model training, the pre-annotation model is jointly trained by the generated data and the labelled data. In the stage used, samples to be annotated are pre-annotated. The CNN network structure of the pre-annotation model is shown in Fig. 6. It is composed of 4 one-dimensional convolutional layers, 4 pooling layers, 2 fully connected layers, and one output layer. Since each generated beat is composed of 95 data points, the real beat for training is downsampled from 286 sampling points to 95. We first divide the extracted beats from the MIT-BIH Arrhythmia Database into a training set and a test set for our experiment. Then the labelled data for input come from the training set, the samples to be pre-annotated come from the test set, and the generated data come from the simulation data generated by the GAN.

Therefore, the intelligent annotation system based on generated data must be supported by a certain amount of labelled data before using the smart function. With the continuous increase in labelled data, the self-learning ability and intelligence will increase.

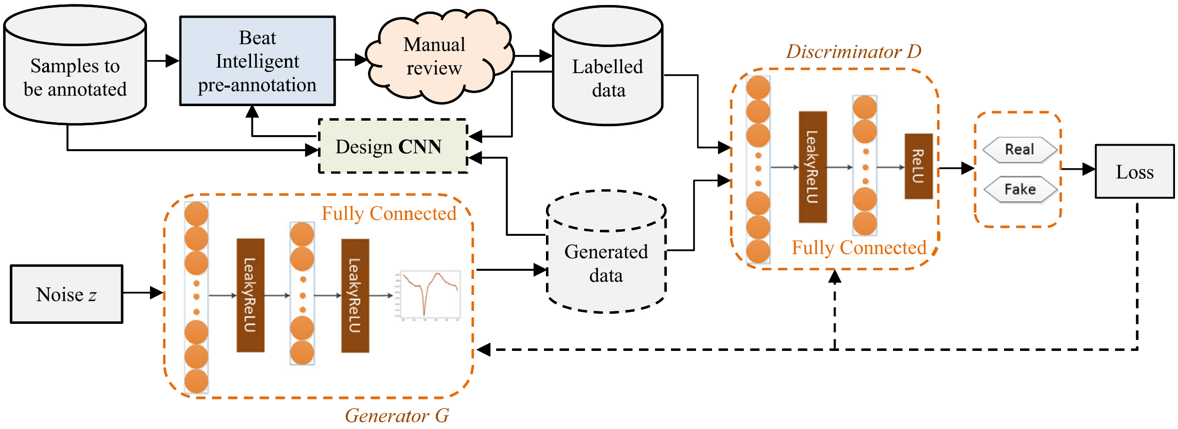


Fig. 5. The flowchart of the beat pre-annotation algorithm based on generated data.

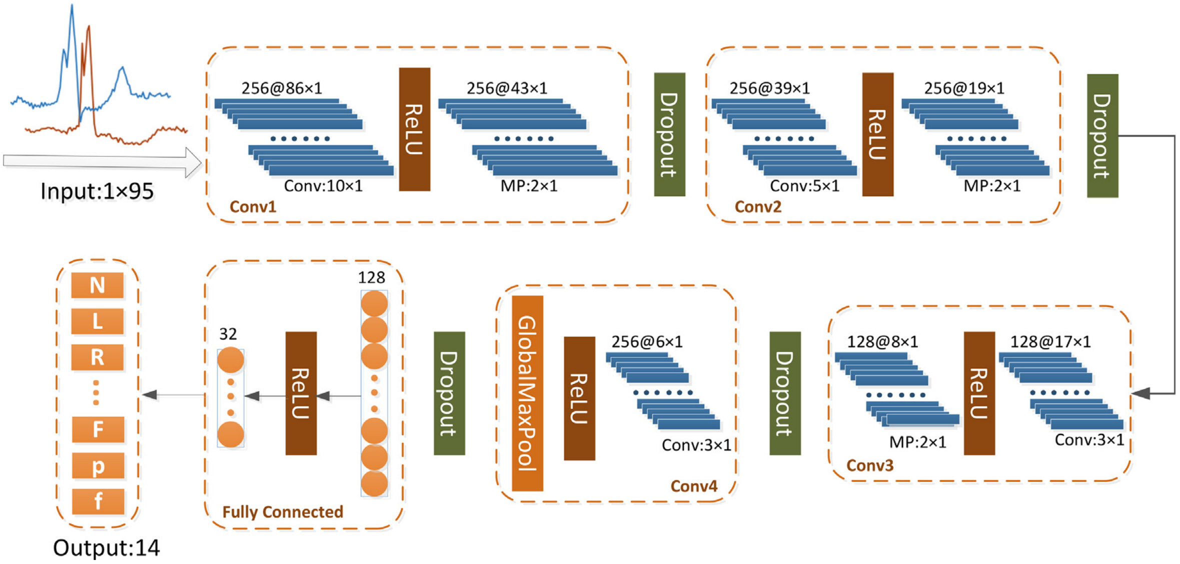


Fig. 6. The structure of the CNN model for beat pre-annotation.

4. Experimental results analysis

The experiment is conducted in three parts. First, we discuss the quality of the generated beats; then, we discuss the performance of the beat pre-annotation model, and finally, we discuss and analyse the efficiency advantages of the intelligent annotation pattern.

4.1. Intelligent simulation data quality assessment

The generated simulation data are used to assist the pre-annotation model's training. In the following experiments, the quality of the simulation data is verified from different perspectives. First, we verify the generation model's validity by the loss curve. Then we discuss the filtering rules of the generated data and the precision of generated data is evaluated with an MMD indicator. Finally, the validity of the generated data is verified by blind judgement.

4.1.1. Validity evaluation of GAN-based generation model

In this work, it is possible to identify whether the training is successful by drawing the loss curves of the generator G and the discriminator D during training, and also verify whether the model is effective. The essence of GAN is generation and confrontation. The G and the D continue to game until it generates data that cannot be distinguished between real and fake. Therefore, G and D's loss curves should constantly be fluctuating in an effective GAN model training. If G drops too fast, it means that D is too weak and can be easily deceived. When D drops too fast, it means that the data generated by G is fake, and it is easy to distinguish from real data. As shown in Fig. 7, where Fig. 7 (a), (b), (c), (d), (e), and (f) are the error curves for generating beats e, j, a, J, S, and E. And there are only 16, 229, 150, 83, 2, and 106 real beats for e, j, a, J, S, and E, respectively. Although the amount of beat is small, it is obvious that G and D's loss curves in all beat generation training are constantly fluctuating. The overall convergence to 1 and the D's loss curve is close to 0.5, indicating that G and D have reached a state of constant gaming.

4.1.2. Accuracy assessment of the generated data

The error curve is used to identify the model's validity. In this subsection, the quality of the generated data is evaluated using the maximum mean discrepancy (MMD) [16] indicator, in which the MMD was initially designed to judge whether two distributions are the same. In this paper, MMD is adopted to measure the similarity between the generated and original data. The definition of the square of the MMD is shown in formula (4). A smaller value of MMD denotes a more remarkable similarity, and thus, this paper prefers a smaller value of MMD.

$$MMD^2 = \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i, y_j) \tag{4}$$

The kernel function selected here is a Gaussian kernel function, a monotonic function of the Euclidean distance of two vectors, as shown in formula (5), in which σ denotes that the bandwidth controls the Gaussian kernel function's local scope.

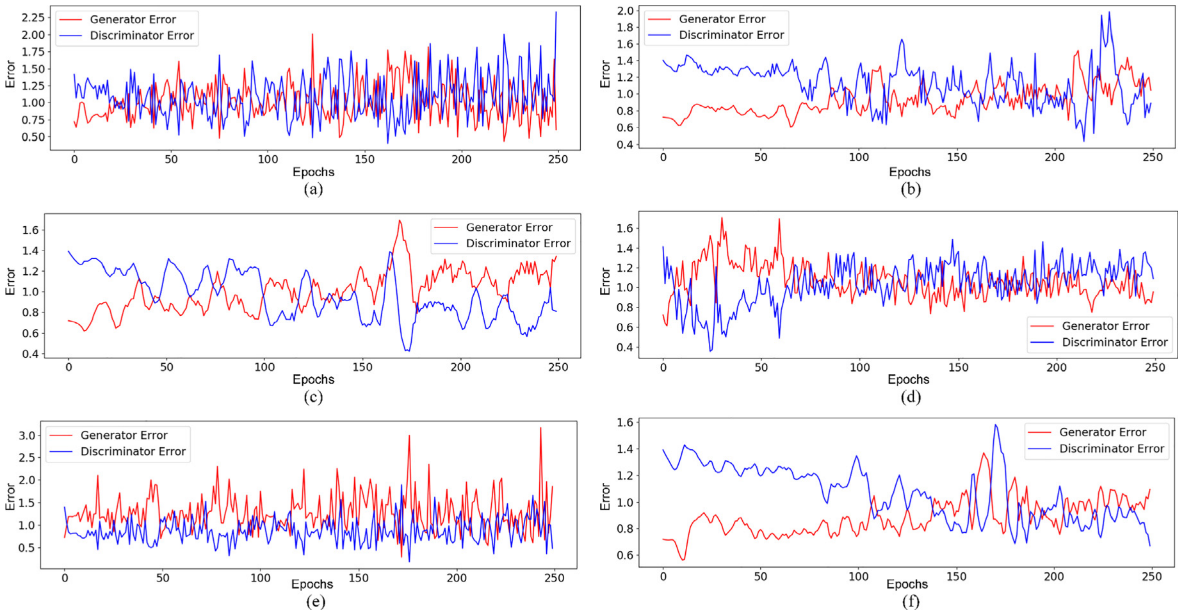


Fig. 7. The error curve graph of GAN-based beat pre-annotation model training. (a), (b), (c), (d), (e), and (f) are GAN errors in training to generate beats e, j, a, J, S, and E, respectively.

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}} \tag{5}$$

We take the fusion of ventricular and normal beat (F) as an example to observe the results and changes in MMD under different parameters and different values. The value of MMD is mainly related to the parameters batch size, noise, and batch. As shown in Fig. 8 (a), (b), and (c), the abscissa axis represents the number of training steps, and the ordinate axis denotes the MMD. The detailed analysis of each parameter is as follows:

Batch size: The number of samples involved in training each time is called the batch size. To objectively observe the changing trend from the batch size, the noise is set to 30 and the batch to 800. It is not difficult to find from Fig. 8(a) that the MMD first reaches the lowest when the batch size is 50. When the batch size values are 100, 160, and 256, the MMD value is approximately 0 and tends to be stable as the number of steps increases.

Noise: When generating the data, the input noise *z* is a set of random numbers drawn from the standard normal distribution and the batch size is set to 50 and the batch to 800. The number of noises is set to 10, 30, 60, and 90. Fig. 8(b) shows that the trend in the MMD values for all noise values are roughly the same, and the smaller the noise is, the faster it reaches the lowest value.

Batch: To facilitate the training, the data set is divided into several small data sets, and the number of samples in the small data set is the batch value. In this experiment, a batch is the number of real beats used in training for each epoch and the noise is set to 30 and the batch size to 50. Fig. 8(c) describes the impact of the batch on the MMD. It can be clearly seen that the larger the batch is, the smaller the MMD, and as the step increases, the MMD with a larger batch value tends to be consistent and stable.

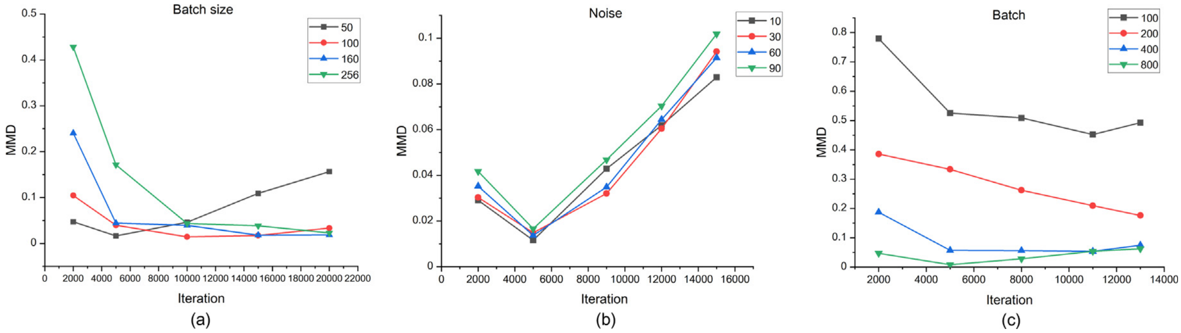


Fig. 8. The analysis of the influence of parameters on MMD by varying the value of batch size, noise, and batch.

To obtain a more comprehensive observation, we set a high iteration value. In general, the MMD value is small with different parameters, and when the relevant parameter values are appropriate, the MMD can easily reach 0.01 and below. Therefore, the generated data is different from the real data, and the difference is minimal. This finding indicates that the generated data and the real data are incredibly similar.

4.1.3. Validity filtering of the generated data

It can be seen from Fig. 7 that the GAN is initially unstable, and thus, the generated data are also unusable. After training for a certain number of iterations, the loss graph reaches convergence, and G and D also reach the game state. At this time, it is difficult for D to distinguish the TRUE and FAKE of the data. Therefore, to filter out invalid generated data, the determination of the first stable convergence point (*Fscp*) is the key. Table 2 describes the method of determining *Fscp* in detail. The comparison of the loss values of G and D's corresponding positions in the window (the length is defined as *LW*) with *Fscp* as the starting point need to be basically balanced, and the degree of balance is defined as the convergence density *Pden*, which is a probability value. The absolute ideal state is that the probability *Pden* that G's loss value is greater than D's loss value (or that D is more significant than G) is exactly 50%. In practical applications, we set *Pden* to a probability range (*Pden*, 1- *Pden*).

To obtain the best *Fscp*, we set *LW* equal to 500, 1000, and 1500 and set the *Pden* between 0.49 and 0.51. The obtained *Fscp* is shown in Table 3.

From the analysis of Table 3, an *LW* of 1000 can ensure that the equilibrium state's density in the window is better than 1500, such as beats R, e, j, A, a, E, p, and f. And *LW* of 1000 can break the limitation of local stability better than 500, such as beat e. Therefore, the final *LW* is set to 1000. When *LW* equals 1000, e needs 3081 iterations to reach a stable convergence state, which is the maximum number of iterations among all types of beats. From the analysis of MMD in Fig. 8, it is easy to determine that as long as the appropriate parameters are set, when the number of iterations of beat F reaches 5000, the difference between the generated data and the real data is already minimal. The MMD value reached a relatively stable state without significant fluctuations. Fig. 9 lists 4 groups of generated beats whose number of iterations does not reach *Fscp* (upper), reached *Fscp* (middle), and whose iterations reached 5000 (bottom). As shown in the figure, the beats whose number of iterations does not reach *Fscp* are not only chaotic in shape and extremely noisy, while the shape of beats that reach *Fscp* is ideal but not smooth enough. In contrast, the beats whose number of iterations reaches 5000 are perfect in shape and smoother. In summary, *Fscp* is set to 5000, and the selected effective generation data are also generated by iterative training starting from *Fscp*.

4.1.4. Availability assessment of generated data

We verified the beat generation model's validity, quantified the difference between the generated data and the real data, and described the validity filtering method of the generated beat. Fig. 10 shows the comparison of 8 groups' beats. The orange signals represent the generated data that reached *Fscp*, and the blue signals denote the real data. It can be observed that the shape of the generated data is extraordinarily close to the real data. The noise input causes a slight difference, and the difference ensures the diversity of the data.

Table 2
The determination of the first stable convergence point for GAN training.

Algorithm: The first stable convergence point (Fscp) determination method.

```
Input:
  -LI: The length of the loss set; the total number of iterations.
  -Glosses=$\{Gloss_i\}_{i=1}^{LI}$: The loss set of generator G (a loss value is obtained after one iteration).
  -Dlosses=$\{Dloss_i\}_{i=1}^{LI}$: The loss set of discriminator D.
  -LW: The length of the window used to calculate the convergence density.
  -Pden: The minimum convergence density value that satisfies the condition in a window of length LW.
Output:
  -Fscp: The first stable convergence point, the number of iterations to reach stable convergence for the first time.
Steps:
  for li = 1:LI
    if Dlosses[li] < Glosses[li]
      Fscp = 0
      for lw = 1:LW
        if Dlosses[li + lw] < Glosses[li + lw]
          Fscp = Fscp + 1
      if Fscp > LW*Pden and Fscp < LW*(1-Pden)
        Obtain Fscp
      end if
      end for
    end if
  end for
```

Table 3
Comparison of *Fscp* with *LW*.

Label	*LW* = 500	*LW* = 1000	*LW* = 1500
R	587	587	441
e	2475	3081	2589
j	1527	1527	1316
A	2327	2004	1618
a	987	732	460
J	252	252	252
S	314	314	314
V	145	145	145
E	1383	1383	1237
F	260	260	260
p	1143	1143	845
f	1218	1218	1016

To further confirm the validity and usability of the generated data, we designed a blind test experiment and invited two senior ECG doctors to participate. One doctor is the director of the electrocardiogram department, and the other is a cardiologist. Both experts are proficient in electrocardiography and have rich experiences in the field of ECG-assisted arrhythmia diagnosis. We prepared 175 beats of all types, including 75 real beats and 100 generated beats. The beat type will be indicated in the experiment, and the doctor will judge whether the beat is real or generated and whether it belongs to the indicated beat type. If the beat is considered to be real and confirmed to be the indicated beat type, it is marked as TRUE. If the beat is considered to be generated or confirmed to be not the indicated beat type, it is marked as FAKE. Table 4 and Table 5 are the confusion matrices generated by the two doctors' judgements. T and F represent the number of beats that are initially TRUE and FAKE, and T' and F' represent the number of TRUE and FAKE beats judged by experts.

The following is a detailed description of the meaning of each cell of the confusion matrix:

Row T and column T': The number of beats that were originally TRUE and judged to be TRUE, represented by TT'.

Row T and column F': The number of beats that were originally TRUE but judged to be FAKE, represented by TF'.

Row F and column T': The number of beats that were originally FAKE but judged to be TRUE, represented by FT'.

Row F and column F': The number of beats that were originally FAKE and judged to be FAKE, represented by FF'.

Table 6 specifically analyses the blind judgement of beats according to Table 4 and Table 5. ACC^d represents the ratio of correct judgements by doctors. TPR^d represents the ratio of true beats that are correctly judged as true. FPR^d represents the ratio of fake beats that are incorrectly judged to be true. The definitions of ACC^d, TPR^d, and FPR^d are as shown in formulas (6), (7), and (8), respectively.

$$ACC^d = \frac{TT' + FF'}{TT' + TF' + FT' + FF'} \tag{6}$$

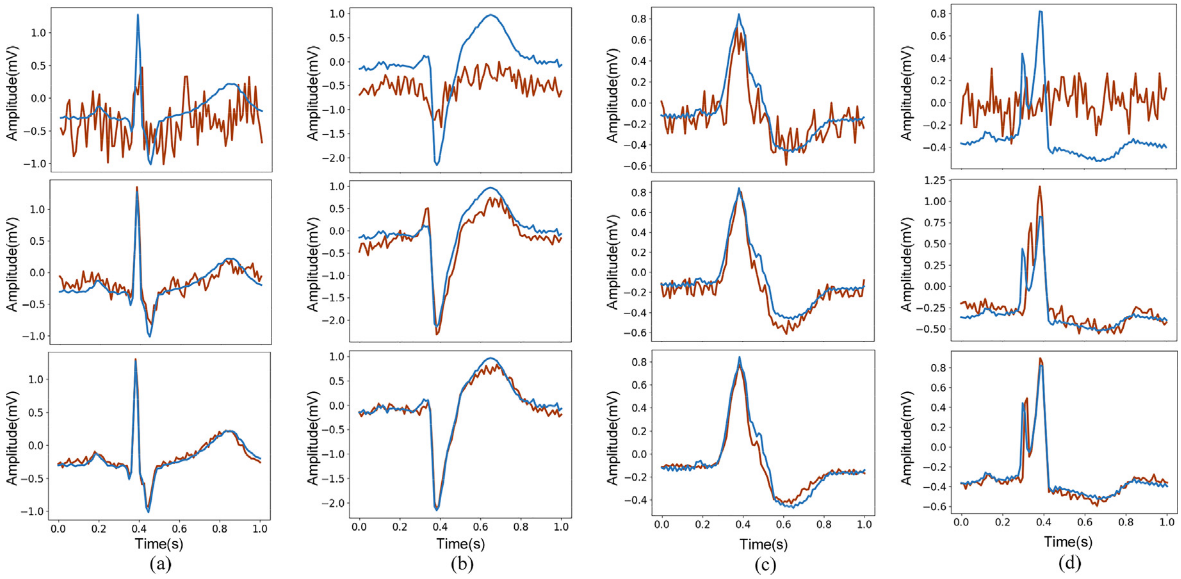


Fig. 9. Comparison of beats in different iteration stages. (a)-(d) are the comparisons of beats R, V, E, and f in different iteration stages. In each group, the upper beats did not reach *Fscp*, the middle reached *Fscp*, and the bottom reached 5000. The orange signals are the generated beats, and the blue signals are the real beats.

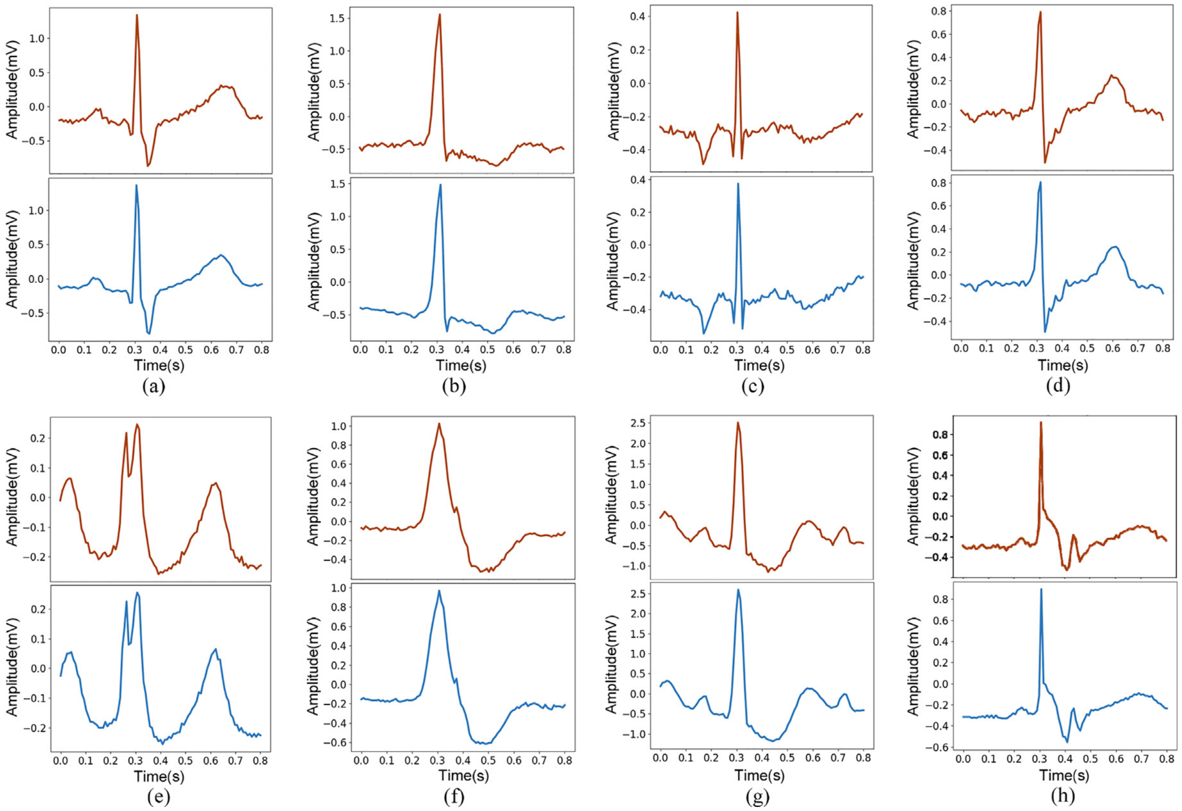


Fig. 10. A comparison of real beats and generated beats. (a)-(h) are the comparison of 8 groups of intelligent simulation beats and real beats. The 8 sets of beats are R, e, j, A, a, E, F and f, the top of each group is the generated intelligent simulation data (orange), and the bottom is the real data (blue).

Table 4
Confusion matrix of doctor 1′s judgement results.

Real	Doctor judgement		
	T'	F'	Total
T	46	29	75
F	76	24	100
Total	122	53	175

Table 5
Confusion matrix of doctor 2′s judgement results.

Real	doctor judgement		
	T'	F'	Total
T	50	25	75
F	75	25	100
Total	125	50	175

Table 6
Analysis of doctors' judgement results.

Doctor	ACC^d	TPR^d	FPR^d
Doc1	40%	61%	76%
Doc2	42.9%	67%	75%

$$TPR^{d} = \frac{TT'}{TT' + TF'} \tag{7}$$

$$FPR^{d} = \frac{FT'}{FT' + FF'} \tag{8}$$

From Table 6, it can be seen that the ACC^{d} of the two doctors was 40% and 42.9%, which are both lower than 50%, which indicates that the doctors could not distinguish whether the data were real or generated. At the same time, it is found that the TPR^{d} of the two doctors are less than the FPR^{d}. In other words, the probability that the generated data are judged to be TRUE is greater than that of the real data. This finding shows that the generated data can be used as a supplement to the real data. Therefore, it can be concluded that the generated data are valid and usable.

4.2. Performance evaluation of beat pre-annotation

To make the performance of beat pre-annotation better, a beat pre-annotation algorithm based on intelligently generated data is proposed. First, the GAN-based generation model is used to generate the data and balance the data set, and then the balanced data set is pre-annotated using a CNN-based pre-annotation model. The performance of beat pre-annotation algorithm is verified from three aspects: (1) The comparison of pre-annotation performance before and after data set balanced; (2) The performance comparison of the proposed model and the classic models; (3) The performance comparison of the proposed method and the state-of-the-art methods.

4.2.1. Performance comparison before and after balancing

It is worth pointing out that the minority class is usually the one with the highest interest from a learning point of view, and it also implies a great cost when it is not well classified [17]. The same is true in clinical medicine. The rarer disease has a higher research value. But in actual classification, fewer cases usually have less effectiveness. Because the standard classification learning algorithms are often biased towards the majority class and have a higher misclassification rate for the minority class [18]. Table 1 describes the distribution of the number of beats in the data set, and it is found that the data set is highly imbalanced. To balance the data set, each class is based on 8000 instances, classes with less than 8000 instances are supplemented with simulation data, while classes with more than 8000 instances are selected 8000 randomly, such as beat N. The number of simulation beats that need to be supplemented for each class are shown in the “Generated beats” column of Table 7. The test set of the balanced and the original data set (unbalanced data set) are the same, both of which are extracted from the real data set with a ratio of 33%.

It can be seen from Table 7 that the accuracy of most of the classes will increase after balancing, especially the minority classes have a large improvement (in bold), such as, beat types with labels e, j, a, J, S, F. It is worth emphasizing that there are only 2 records in class S, and the accuracy is 0% without balancing and 100% after balancing. Class e has only 16 records, and the accuracy after balancing is 20% higher than before balancing. In addition, the accuracies of other beats have increased or stayed the same basically after balancing.

To further verify whether *Fscp* is effective and accurate, use the generated data before *Fscp*, after *Fscp*, and the mixed (before and after *Fscp*) to balance the data set. The pre-annotation performances are shown in Table 7. By comparison, the per-

Table 7
The performance comparison of before and after balancing.

Label	Real beats	Generated beats	Test data set (33% of real data)	ACC^{c} (Imbalanced data set)%	ACC^{c} (Balanced with the data before *Fscp*)%	ACC^{c} (Balanced with mixed data)%	ACC^{c} (Balanced with the data after *Fscp*)%
N	8000	0	2640	99.85	99.55	99.62	99.85
L	8072	0	2664	99.85	99.66	99.62	99.85
R	7255	745	2394	99.87	99.67	99.67	99.79
e	**16**	7984	5	**60**	0	40	**80**
j	**229**	7771	76	**94.74**	90.79	96.05	**98.68**
A	2546	5456	840	96.55	95.95	97.50	97.50
a	**150**	7850	49	**79.59**	75.51	85.71	**89.80**
J	**83**	7917	27	**88.89**	88.89	92.59	**92.59**
S	**2**	7998	1	**0**	100	100	**100**
V	7129	871	2353	98.85	98.56	98.17	98.60
E	106	7894	35	97.14	97.14	94.29	97.14
F	802	7198	265	**92.83**	90.19	93.21	**95.09**
p	7024	976	2318	99.87	99.91	99.83	99.91
f	982	7018	324	96.91	97.53	98.77	97.22
Total	42,396	69,678	13,991	99.14	98.85	99.04	99.28

Annotation: ACC^{c}(Imbalanced data set) → The accuracy of the test set, the training set is from ”Real beats” and the test set is from ”Test data set (33% of real data)”. ACC^{c}(Balanced with the data after *Fscp*) → The accuracy of the test set, the training set is from ”Real beats”+“Generated beats”(generated by iterative training starting from *Fscp*) and the test set is from ”Test data set (33% of real data)”, and so on.

formances before *Fscp* are lower than that of after *Fscp* and the mixed. Simultaneously, it can be found that, for most classes, the performances of using the before *Fscp* and the mixed generated data to balance the data set are lower than that of the unbalanced data set. So the filtering of the validity of the generated data is vital, and invalid data will directly affect the pre-annotation performance.

Fig. 11(a) and Fig. 11(b) are the confusion matrix before and after the data set's balanced (Balanced with the generated data after *Fscp*). By comparison, it can be found that the dark distribution in Fig. 11(b) is more concentrated on the diagonal. Obviously, the performance of the confusion matrix Fig. 11(b) is better than confusion matrix Fig. 11(a), and the performances difference between before and after balancing for the majority classes are tiny or even the same, while for the minority classes, the performance after balancing are greatly improved. Finally, the total accuracy after balancing is 99.28%, which is higher than the 99.14% before balancing. Therefore, it can be verified that the beat pre-annotation model based on simulation data can not only deal with the problem of the high misclassification rate of minority class but also improve the overall classification performance. At the same time, it can be found that the accuracy before balancing is quite good, which shows that the overall performance of the CNN-based beat pre-annotation model proposed in this paper is quite good even for the imbalanced data set.

4.2.2. Performance evaluation of beat pre-annotation model

To verify the performance of the CNN-based beat pre-annotation model proposed in this paper, we compare it with multiple classical models in weka. The training set and test set of all models are the same. The training set is balanced by simulation data after *Fscp*, and the test set is drawn from the real data by an equal proportion of 33% as shown in Table 7.

Kappa coefficient [19] is a method of checking consistency in statistics, and it is often used to evaluate the accuracy of a multi-class classification model. Kappa coefficient is a ratio, and the calculation is based on the confusion matrix, which represents the ratio of error reduction between classification and completely random classification. The calculation of Kappa coefficient *kap* is shown in formulas (9) and (10).

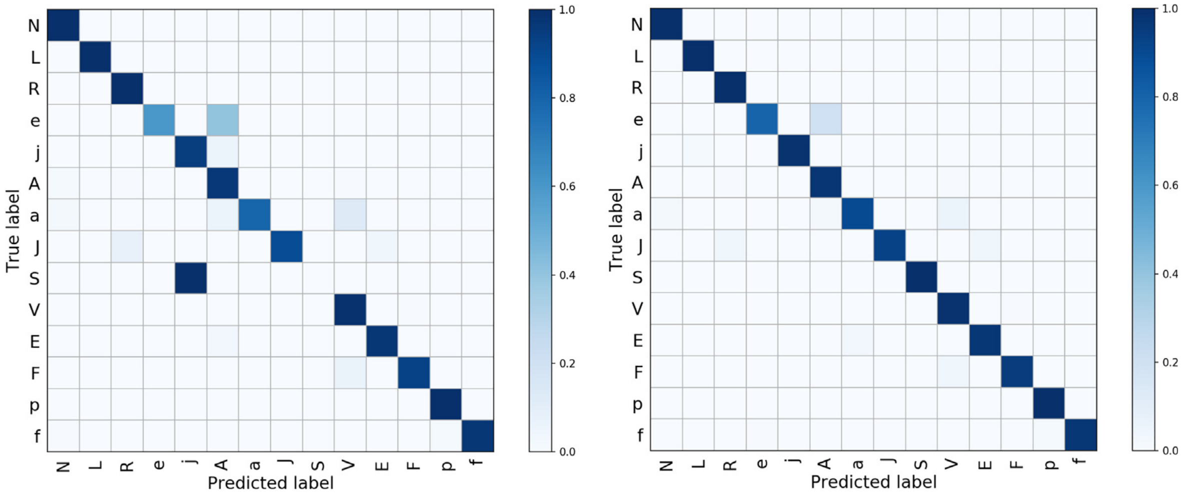
$$kap = \frac{p_0 - p_e}{1 - p_e} \tag{9}$$

where p_o represents the overall classification accuracy which is defined as the number of samples correctly classified divided by the number of overall samples. The calculation of p_e is shown in formula (10):

$$p_e = \frac{a_1 * b_1 + a_2 * b_2 + \cdots + a_C * b_C}{n * n} \tag{10}$$

where,$a_1, a_2, \cdots, a_C$ represent the number of real samples of each class. $b_1, b_2, \cdots, b_C$ represent the number of predicted samples of each class, and n represents the total number of samples.

The range of Kappa coefficient is [0,1] in practical applications, and the higher the value, the higher the classification performance of the model. Generally, Kappa coefficients can be divided into five groups to represent the different consistency levels [20], which are 0.0 $\sim$ 0.20: slight, 0.21 $\sim$ 0.40: fair, 0.41 $\sim$ 0.60: moderate, 0.61 $\sim$ 0.80: substantial, 0.81 $\sim$ 1: almost perfect. To comprehensively evaluate the model, we use accuracy (ACC^c %) and Kappa coefficient(kap %) to measure the performance of the model. The results are shown in Table 8. The Kappa coefficients show that all models except NaiveBayes,



(a) Imbalanced data set

(b) Balanced data set

Fig. 11. Confusion matrix. (a) is the confusion matrix of the original data set, (b) is the confusion matrix of the data set after balancing with intelligent simulation data.

Table 8
The performance comparison with classical models.

Model	ACC^c(%)	kap(%)
NaiveBayes[21]	54.41	47.92
BayesNet[22]	67.99	61.27
J48[23]	93.55	92.33
Bagging[24]	94.82	93.84
Dagging[25]	76.69	73.1
RandomCommittee[26]	96.95	96.37
PART[27]	94.83	93.84
KStar[28]	96.85	96.25
MLP[29]	89.69	87.8
RandomForest[30]	96.70	96.08
IB1[31]	97.88	97.47
RandomTree[30]	91.91	90.4
RotationForest[32]	98.18	97.84
This paper	**99.28**	**99.14**

BayesNet, and Dagging have reached the perfect level. And our model has not only the highest Kappa coefficient but also the highest accuracy. It can be concluded that the CNN-based beat pre-annotation model designed in this paper has significant advantages in performance.

4.2.3. Comparison with previous methods

Besides, we compare our method with the state-of-the-art researches that were published in recent years, which all use the MIT-BIH Arrhythmia Database. The comparison results are shown in Table 9, which details classes number and classification method. The methods range from 2 classes to 17 classes. The previous methods classify beats only or classify beats and rhythms for arrhythmia detection. In the proposed intelligent annotation system, Ventricular flutter is defined as a rhythm, and the "ECG beats" column in Table 9 begins with "(" as rhythm labels. We summarize the classification methods into the following 5 categories:

1) Divide beats into normal and abnormal, as in [39];
2) Divide beats according to the Advancement of Medical Instrumentation standard [45], such as [36,40,43];
3) Choose common beat types, such as [34,35] and [42];
4) Select common beats and common rhythms together for arrhythmia detection, such as [41];
5) A detailed and complete division of beats, such as [33,37,38,44] and our work.

In machine learning, the more classes there are on the same data set, the less likely it is to guarantee performance. The comparison in Table 9 shows that the performance of the proposed model for 14 classes is not only higher than the models with finer granularity, such as [37,38,41,44], and it is also higher than the models with a coarser granularity. Therefore, it can be concluded that the CNN-based beat pre-annotation model of this paper achieved the best performance.

Table 9
Performance comparison of the proposed method and other high-performance approaches on the MIT-BIH Arrhythmia Database.

Authors	Year	ECG beats	Number of beat types	ACC^c(%)
Rodriguez J et al.[33]	2005	N, L, R, j, A, a, J, S, V, E, F, p, f, (VFL	14	96.13
Melgani F et al.[34]	2008	N, L, R, A, V, p	6	91.67
Yu S N et al.[35]	2008	N, L, R, A, V, E, p, (VFL	8	98.71
Zhang Z et al.[36]	2014	BN(N,L,R) BS(e,j,A,a,J,S) BV(E,V) BF(F)	4	86.66
Raj S et al.[37]	2016	N, L, R, e, j, A, a, J, V, E, F, p, f, Q, x, (VFL	16	99.18
Chen S et al.[38]	2017	N, L, R, e, j, A, a, J, V, E, F, f, Q, x, (VFL	15	98.46
Sannino G et al.[39]	2018	N, AN	2	99.09
Li W et al.[40]	2018	BN(N,L,R,e,j) BS(a,A,S,J) BV(E,V) BF(F) BQ(Q, p, f)	5	99.01
Pławiak P et al.[41]	2018	N, L, R, A, V, F, p, (AFL, (AFIB, (SVTA, (PREX, (B, (T, (VT, (IVR, (VFL, (BII	17	90
Huang J et al.[42]	2019	N, L, R, A, V	5	99.0
Wang H et al.[43]	2020	BN(N,L,R,e,j) BS(a,A,S,J) BV(E,V) BF(F) BQ(Q, p, f)	5	99.06
Yang H et al.[44]	2020	N, L, R, e, j, A, a, J, S, V, E, F, p, f, Q	15	97.7
This paper		N, L, R, e, j, A, a, J, S, V, E, F, p, f	14	**99.28**

Annotation: AN → Abnormal beats; Q → Unclassifiable beat; x → Non-conducted P-wave(blocked APB); (AFL → Atrial flutter; (AFIB → Atrial fibrillation; (SVTA → Supraventricular tachyarrhythmia; (PREX → Pre-excitation(WPW); (B → Ventricular bigeminy; (T → Ventricular trigeminy; (VT → Ventricular tachycardia; (IVR → Idioventricular rhythm; (VFL → Ventricular flutter; (BII → 2° heart block. For the meaning of other labels, refer to the "label" in Table 1 and its corresponding "description". BN, BS, BV, BF, and BQ are big categories of beats, and the specific beat types included are described in detail in "()".

4.3. Performance comparison of three annotation patterns

We design a set of experiments to evaluate the intelligent annotation system's performance compared with the traditional full manual annotation pattern and semi-automatic annotation pattern. To make a comprehensive and fair comparison, 4 electrocardiology experts with different ages, medical qualifications, and physiological characteristics are invited to participate, which shows in Table 10. The experts are from the top three hospitals in Henan province, which has a population of 109 million. The range of the age of 4 experts is between 33 and 59. The title also covers the attending physician, deputy chief physician, and chief physician. We collected 9 real ECG records and divided them into 3 groups, which are the full manual annotation group, semi-automatic annotation group, and audit group. At the same time, according to the operation complexity of the annotation, each group contains a low-complexity, an intermediate-complexity, and a high-complexity ECG records. Here, records 1, 4, and 7 are low complexity, records 2, 5, and 8 are intermediate complexity and records 3, 6, and 9 are high complexity, respectively.

The specific operation of each annotation pattern is described as follows:

(1) Full manual annotation: The full manual annotation pattern requires experts to locate the beat and select the type manually, which is the most original pattern.
(2) Semi-automatic annotation: The semi-automatic annotation pattern will automatically locate the beat and mark it as normal(N) by default. The annotation expert only needs to change the type of the beat that is not N.
(3) Intelligent annotation: The intelligent annotation pattern proposed by this paper consists of two parts, the first is to inspect the pre-annotation results, and the second is to modify the beats that have been wrongly pre-annotated. Therefore, the time used in the intelligent annotation pattern is the audit time (all beats) plus the modification time (the beats that are wrongly pre-annotated). So, we set the audit group, the 3 records in the group have been pre-annotated accurately, and the experts don't know whether there are errors in the pre-annotation results. Then we get pure audit time.

We define the annotation time of 10 beats as a unit annotation time (UAT_{10b}), which is shown in formula (11):

$$UAT_{10b} = \frac{AT_{experts}}{N_{beats}} * 10 \tag{11}$$

where $AT_{experts}$ denotes the average time of each record annotated by each expert and N_{beats} represents the number of beats contained in the record. To obtain a more fair comparison of the experimental, we also compare the average unit annotation time $AUAT_{10b}$ of 3 records for each annotation pattern.

Table 10
Description of experts' information.

Experts	Age	Education background	Professional title	Years of employment (year)
1	59	B.S.Med	Chief Physician	38
2	33	M.S.Med	Attending physician	6
3	34	M.S.Med	Attending physician	6
4	51	M.S.Med	Deputy chief physician	28

Annotation: B.S.Med → Bachelor of Science in Medicine; M.S.Med → Master of Science in Medicine.

Table 11
Quantitative analysis of annotation time for different annotation patterns.

Annotation pattern	Full manual			Semi-automatic			Audit		
ECG record	1	2	3	4	5	6	7	8	9
Types of arrhythmias	N	R	p	N	V	p	N	L	p
N_{beats}/duration	18b/16s	26b/21s	24b/18s	16b/13s	35b/21s	30b/26s	23b/17s	37b/28s	36b/27s
Expert 1	50s	61s	137s	13s	20s	91s	10s	23s	10s
Expert 2	45s	51s	103s	10s	18s	88s	7s	10s	9s
Expert 3	43s	54s	94s	9s	11s	85s	6s	7s	10s
Expert 4	45s	55s	112s	11s	19s	98s	8s	9s	7s
$AT_{experts}$	45.75s	55.25s	111.5s	10.75s	17s	90.5s	7.75s	12.25s	9s
UAT_{10b}	25.42s	21.25s	46.46s	6.72s	4.86s	**30.17**s	3.37s	3.31s	2.50s
$AUAT_{10b}$	**31.04s**			**13.92s**			**3.06s**		
Intelligent annotation(10b)	3.06s + **30.17s***(1-*ACC*^{c}) = **3.28s**								

Annotation: N_{beats}/duration → Number of beats/duration of ECG record, such as 18b/16 s means that this is a 16 s record with 18 beats; $AT_{experts}$ → Average time of 4 experts to annotate the corresponding record; UAT_{10b} → Unit(10 beats) annotation time, calculated by $AT_{experts}$; $AUAT_{10b}$ → Average unit time of the pattern, calculated by UAT_{10b}; ACC^c = 99.28%.

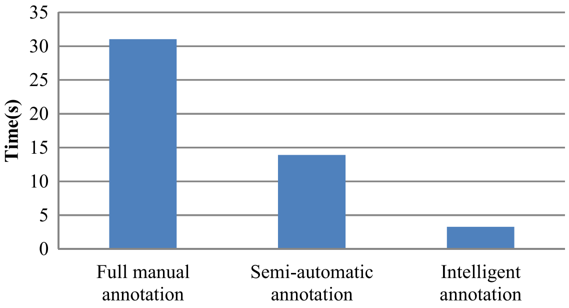


Fig.12. Comparison of beat annotation time of three annotation patterns (10 beats).

According to the above calculation, the $AUAT_{10b}$ of the full manual pattern and the semi-automatic pattern can be obtained. The time of the intelligent annotation is the audit time of all beats adding the modification time of correcting the wrongly pre-annotated beats (the error proportion is 1- ACC^c). The modification time is referenced from the UAT_{10b} of record 6. Since its type is "p", all beats should be changed from "N" to "p", the pure modification time can be obtained. The calculation and result of the time used for intelligent annotation are shown in the last row of Table 11.

From Table 11, we can conclude that the ECG record containing 10 beats requires 31.04 s in full manual annotation pattern, 13.92 s in semi-automatic annotation pattern, and 3.28 s in intelligent annotation pattern, as shown in Fig. 12. According to these analyses, it can be concluded that intelligent annotation can reduce working time by 89.43% compared with the full manual pattern, and can reduce working time by 76.44% compared with the semi-automatic pattern.

5. Conclusions

This paper introduces a human–machine integration ECG intelligent annotation system based on simulation data generation. The system can change the traditional manual annotation pattern and incorporate a variety of intelligent technologies to assist the ECG annotation in such a way that the work of annotation experts can be transformed from annotation to audit, which can significantly improve the annotation efficiency. However, all of the intelligent technologies are based on labelled data. Therefore, to solve the problem of insufficient labelled data, the concept of generating accurate simulation data to assist model training is proposed, including the intelligent screening model of samples to be labelled and the pre-annotation models. Beat annotation is the basis of ECG annotation, and thus, we propose a beat pre-annotation model based on intelligent simulation beat generation. First, a GAN model for generating accurate beats was designed. After various verifications, it was verified that the model could generate specific types of beats that can supplement labelled data. Then, a 14-class beat pre-annotation model based on a CNN is proposed. The model is jointly trained by generated data and real data, and the accuracy obtained on the test set is 99.28%. This model's performance is higher than that of the existing state-of-the-art methods. The most far-reaching significance lies in the annotation. Intelligent annotation can reduce the working time by 89.43% compared with a full manual pattern and reduce the working time by 76.44% compared with a semi-automatic pattern.

The future work arising from this paper can focus on the following aspects: (1) It can be extended to study the rhythm pre-annotation model based on precise, intelligent simulation rhythm data, in other words, to propose a model that can generate specific types of ECG rhythm segments and a high-performance rhythm pre-annotation model; (2) It is necessary to design a high-performance, multi-label classification model for conclusive pre-annotation; (3) It is also necessary to design an excellent quality inspection model to assist in the filtering of samples to be annotated; (4) It is necessary to continuously supplement and improve the core algorithms required by the ECG intelligent annotation system. The concept of intelligent annotation can also be extended to other fields that require professional annotation, especially various research directions in computer-aided diagnosis. In conclusion, solving the difficulty of ECG signals annotation has far-reaching significance for medical intelligence research.

CRediT authorship contribution statement

Haiyan Wang: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Yanjie Zhou:** Conceptualization, Validation, Formal analysis, Writing – review & editing, Project administration. **Bing Zhou:** Software, Visualization, Investigation, Resources. **Xiangdong Niu:** Software, Visualization. **Hua Zhang:** Data curation, Investigation. **Zongmin Wang:** Conceptualization, Software, Supervision, Funding acquisition.

Declaration of Competing Interest

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

Acknowledgments

This research was supported by the following grants: The Key Research, Development, and Dissemination Program of Henan Province (Science and Technology for the People). (No. 182207310002).

References

[1] G.B. Moody, WAVE(Version 6.12)[Software].Available from https://archive.physionet.org/physiotools/wug/wu g.htm, 2019.

[2] G.B. Moody, R.G. Mark, The impact of the MIT-BIH Arrhythmia Database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50.

[3] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation. 101 (23) (2000) e215–e220.

[4] J.W. Zhang, X. Liu, J. Dong, Ccdd: an enhanced standard ecg database with its management and annotation tools, Int. J. Artif. Intell. Tools. 21 (05) (2012) 1240020.

[5] P.E. McSharry, G.D. Clifford, L. Tarassenko, L.A. Smith, A dynamical model for generating synthetic electrocardiogram signals, IEEE Trans. Biomed. Eng. 50 (3) (2003) 289–294.

[6] Z. Li, M. Ma, ECG modeling with DFG, in: IEEE Engineering in Medicine & Biology Conference, IEEE, 2005, pp.2691-2694.

[7] O. Sayadi, M.B. Shamsollahi, G.D. Clifford, Synthetic ECG generation and bayesian filtering using a gaussian wave-based dynamical model, Physiol. Meas. 31 (10) (2010) 1309–1329.

[8] E.K. Roonizi, R. Sameni, Morphological modeling of cardiac signals based on signal decomposition, Comput. Biol. Med. 43 (10) (2013) 1453–1461.

[9] T. Golany, K. Radinsky, PGANs: Personalized generative adversarial networks for ECG synthesis to improve patient-specific deep ECG classification, in: The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 2019, pp. 557-564.

[10] F. Zhu, F. Ye, Y. Fu, Q. Liu, B. Shen, Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network, Sci. Rep. 9 (1) (2019) 1–11.

[11] A. Hernandez-Matamoros, H. Fujita, H. Perez-Meana, A novel approach to create synthetic biomedical signals using BiRNN, Inform. Sci. 541 (2020) 218–241.

[12] H.Y. Wang, Z.Y. Ge, Z.M. Wang, Accurate ECG data generation with a simple generative adversarial network, in: 2020 2nd International Conference on Artificial Intelligence and Computer Science(AICS), IOP, 2020, pp. 1-5.

[13] G. Wang, C. Zhang, Y. Liu, H. Yang, D. Fu, H. Wang, P. Zhang, A global and updatable ECG beat classification system based on recurrent neural networks and active learning, Inform. Sciences 501 (2019) 523–542.

[14] H. Wang, H. Dai, Y. Zhou, B. Zhou, P. Lu, H. Zhang, Z. Wang, An effective feature extraction method based on GDS for atrial fibrillation detection, J. Biomed. Inform. 119 (2021) 103819, https://doi.org/10.1016/j.jbi.2021.103819.

[15] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, MIT press Cambridge, 2016.

[16] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Scholkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) e49–e57.

[17] Z. Chen, J. Duan, L.i. Kang, G. Qiu, A hybrid data-level ensemble to enable learning from highly imbalanced dataset, Inform. Sci. 554 (2021) 157–176.

[18] R.M. Pereira, Y.M. Costa, C.N. Silla Jr, Toward hierarchical classification of imbalanced data using random resampling algorithms, Inform. Sci. 578 (2021) 344–363.

[19] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46.

[20] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1) (1977) 159–174.

[21] G.H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[22] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (2–3) (1997) 131–163.

[23] J.R. Quinlan, C4.5: Programs for machine Learning, Morgan Kaufmann Publishers Inc., ISBN:978-1-55860-238-0, 1993.

[24] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[25] K.M. Ting, I.H. Witten, Stacking bagged and dagged models. In: Fourteenth international Conference on Machine Learning, San Francisco, CA, 1997, pp.367–375.

[26] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques(third edition), Elsevier, 2011.

[27] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: The Fifteenth International Conference on Machine Learning, 1998, pp. 144–151.

[28] J.G. Cleary, L.E. Trigg, et al., K*: An instance-based learner using an entropic distance measure, in: Proceedings of the 12th International Conference on Machine Learning, Vol. 5, 1995, pp. 108–114.

[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.

[30] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[31] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[32] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.

[33] J. Rodriguez, A. Goni, A. Illarramendi, Real-time classification of ECGs on a PDA, IEEE Trans. Inf. Technol. Biomed. 9 (1) (2005) 23–34.

[34] F. Melgani, Y. Bazi, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, IEEE Trans. Inf. Technol. Biomed. 12 (5) (2008) 667–677.

[35] S. YU, K. CHOU, Integration of independent component analysis and neural networks for ECG beat classification, Expert Syst. Appl. 34 (4) (2008) 2841–2846.

[36] Z. Zhang, J. Dong, X. Luo, K.S. Choi, X. Wu, Heartbeat classification using disease-specific feature selection, Comput. Biol. Med. 46 (1) (2014) 79–89.

[37] S. Raj, K.C. Ray, O. Shankar, Cardiac arrhythmia beat classification using DOST and PSO tuned SVM, Comput. Meth. Programs Biomed. 136 (2016) 163–177.

[38] S. Chen, W. Hua, Z. Li, J. Li, X. Gao, Heartbeat classification using projected and dynamic features of ECG signal, Biomed. Signal Process. Control 31 (2017) 165–173.

[39] G. Sannino, G.D. Pietro, A deep learning approach for ECG-based heartbeat classification for arrhythmia detection, Future Gener. Comp. Sy. 86 (2018) 446–455.

[40] W. Li, J. Li, Local Deep field for electrocardiogram beat classification, IEEE Sens. J. 18 (4) (2018) 1656–1664.

[41] P. Pławiak, Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system, Expert Syst. Appl. 92 (2018) 334–349.

[42] J. Huang, B. Chen, B. Yao, W. He, ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network, IEEE Access 7 (2019) 92871–92880.

[43] H. Wang, H. Shi, X. Chen, L. Zhao, Y. Huang, C. Liu, An improved convolutional neural network based approach for automated heartbeat classification, J. Med. Syst. 44 (2) (2020) 1–9.

[44] H. Yang, Z. Wei, Arrhythmia recognition and classification using combined parametric and visual pattern features of ECG morphology, IEEE Access 8 (2020) 47103–47117.

[45] ANSI/AAMI, Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms, Association for the Advancement of Medical Instrumentation (AAMI), 2008, American National Standards Institute, Inc. (ANSI), 2008 ANSI/AAMI/ISO EC57, 1998-(R).