Expert Systems With Applications 261 (2025) 125482

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A Low-Cost and stable DAL arrhythmia detection algorithm based on the weak stratification query strategy of morphological statistical features

Haiyan Wang [a,b], Yanjie Zhou [c], Xiangdong Niu [d], Daijun Liu [a], Lingling Li [a,*], Ying Duan [a], Zongmin Wang [b]

[a] School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450046, China
[b] Collaborative Innovation Centre for Internet Healthcare, Zhengzhou University, Zhengzhou 450052, China
[c] School of Management, Zhengzhou University, Zhengzhou 450001, China
[d] The Fifth Affiliated Hospital of Zhengzhou University, Zhengzhou University, Zhengzhou 450000, China

ARTICLE INFO

Keywords:
Arrhythmia Detection
Deep active learning
Weak Stratification of Morphological Statistical Features
Query Strategy
CNN

ABSTRACT

The annotation cost of electrocardiogram (ECG) data is extremely high, resulting in a lack of labeled data. However, most existing models are based on supervised learning and highly dependent on labeled data. Therefore, this study proposes a low-cost and stable arrhythmia detection algorithm based on deep active learning to reduce annotation costs and develop a model with a low dependence on labeled data. The algorithm first proposes a Skew series query strategy based on the weak stratification of morphological statistical features, especially for ECG data, including Skew, Skewierste, Skewier-C, and Skewier-C/2, and develops a CNN-based classifier. Finally, experiments verified that the query strategy proposed in this sstudy has higher stability and adaptability than other classical ECG strategies, and that the performance of the proposed CNN is also higher than that of other classical classifiers. The heartbeat detection algorithm based on deep active learning (DAL) proposed in this study can significantly reduce the dependence on labeled data, significantly reducing annotation costs.

1. Introduction

Arrhythmia refers to an abnormal heart beat, rhythm, and frequency, some diseases can cause sudden death if not treated in time, which is very harmful. Therefore, early diagnosis and appropriate treatment are required. Electrocardiogram (ECG) is the most important examination method for diagnosing arrhythmia because it is fast, convenient, and non-invasive (Wang et al., 2023). With the rapid increase in the number of arrhythmia patients, research and application requirements for computer-based arrhythmia-assisted diagnosis are increasing. The effectiveness of computer-aided diagnosis is determined by the performance of the diagnostic model, which is determined by the scale and quality of the labeled data.

Although massive amounts of ECG data are generated daily, they remain unlabeled. ECG annotation is different from general target detection, which only needs to mark whether the animal in the picture is a “cat” or a “dog,” ECG needs to develop professional annotation software and be annotated by ECG experts, which is a cumbersome process and requires high professionalism. According to research needs, the annotation content of an ECG includes heartbeat, rhythm, morphology, and conclusive diagnosis (Wang et al., 2021), which requires precise positioning and accurate mark. It is a 6 s-long ECG segment that includes beat, rhythm, and morphology annotation, as shown in Fig. 1. It can be observed that much content that needs to be located and annotated. Professional and cumbersome ECG annotation makes the annotation cost higher than that of general target detection, leading to a lack of ECG labeled data. This hinders the development of computer-aided intelligent ECG diagnosis to a certain extent; therefore, reducing the annotation cost is an urgent problem to be solved.

Artificial intelligence detection methods in the field of ECG can be divided into two categories. One is the traditional method based on feature extraction, the most widely used is the RR interval (RRI), such as Chazal et al. (2004), Llamedo and Martínez (2010), and Raj et al. (2016) detected heartbeats based on RRI; Andersen et al. (2019), Li et al. (2017), and Zhou et al. (2014) detected atrial fibrillation based on RRI; and Chen et al. (2014) detected obstructive sleep apnea (OSA) based on RRI by using SVM. For this purpose, wavelet-based methods are typically used. Kim et al. (2011) divided heartbeats into five categories

* Corresponding author.
E-mail address: lilingling@zua.edu.cn (L. Li).

https://doi.org/10.1016/j.eswa.2024.125482
Received 28 August 2022; Received in revised form 27 August 2023; Accepted 29 September 2024
Available online 30 September 2024

according to the AAMI standard(ANSI/AAMI, 2008), proposed a heartbeat detection method based on continuous wavelet transform (CWT) using extreme learning machine (ELM) classifier, and Asgari et al. (2015) proposed an atrial fibrillation detection method based on stationary wavelet transform (SWT) using SVM. There are also some other methods, such as Bozkurt et al. (2020) using KNN, SVM, DT, and ensemble classifiers for detecting OSA based on Heart Rate Variable (HRV); Wang et al. (2021) proposed an atrial fibrillation detection algorithm based on gradient set(GDS); Thakor et al. (1990) proposed a ventricular tachycardia and fibrillation detection method based on threshold crossing interval(TCI); Yin et al. (2020) proposed an ECG signal discernment method based on entropy. Geweid and Chen (2022) detected atrial fibrillation from short single-lead ECG recordings using a hybrid approach of dual support vector machine (HA-DSVM) based on a combination of different features.

Another is the point-to-point detection method based on deep learning (DL), the most commonly used is CNN, such as Kiranyaz et al. (2015) and Acharya et al. (2017) respectively used CNN to detect arrhythmia based on beats, Acharya et al. (2017) used CNN to detect myocardial infarction, Attia et al. (2019) used CNN to predict atrial fibrillation, Yao et al. (2020) and Ge et al. (2021) used CNN to detect 9 common arrhythmia diseases and ECG signal characteristics. RNN is also commonly used, such as Saadatnejad et al. (2019) used LSTM to detect heartbeats, Tan et al. (2018) used LSTM and CNN to detect myocardial infarction and congestive heart failure, Li et al. (2020) explored multimodal emotion recognition using LSTM and RNN, Erdenebayar et al. (2019) used gated-recurrent unit (GRU) and CNN to detect sleep apnea. Other deep learning techniques have also been applied to artificial intelligence detection of ECG, such as Wang et al. (2021) used DNN to detect atrial fibrillation, Ribeiro et al. (2020) used DNN to detect 6 common arrhythmia diseases and Taji et al. (2017) detected Atrial fibrillation by using the Deep belief network (DBN).

Most ECG detection algorithms are based on supervised learning (Berkaya et al., 2018; Pławiak, 2018), and the model performance is completely dependent on labeled data, especially in deep learning (Liu et al., 2021), where excellent detection performance comes from a high degree of greed for labeled data; therefore, the cost of model training is high. Repeated annotation of samples with the same interestingness had little effect on model improvement, but invisibly increased the annotation cost. Therefore, reducing invalid and redundant annotations and developing models with low dependence on labeled data significantly reduces annotation costs. Some researchers have proposed building semi-supervised (Zhai et al., 2020), unsupervised (Jovic & Bogunovic, 2011) and active learning models, in which active learning attempts to train models with a small amount of labeled data and achieve the performance obtained by training a large amount of labeled data to relieve the pressure caused by the lack of labeled data. Active learning has been widely used in image processing (Joshi et al., 2009), and is undoubtedly a good choice.

Active learning (AL) can actively select the most valuable unlabeled samples for annotation with the goal of reducing redundancy and repeated annotation, and it is expected to achieve the anticipated performance of the model with a small number of labeled samples. Pasolli and Melgani (2010) applied active learning to ECG intelligence detection earlier and proposed three query strategies, (1) Margin Sampling (MS), (2) Posterior Probability Sampling (PPS), and (3) Query by Committee (QBC). By classifying six common heartbeats on SVM, it was proven that active learning can significantly reduce the need for labeled data while ensuring good performance. With the rapid development of deep learning, some researchers have applied AL to dynamically update the training of deep learning models to improve their performance (Rahhal et al., 2016; Shi et al., 2020; Wang et al., 2019). Heartbeat detection is the most commonly used detection method. For example, beats are divided into five (Wang et al., 2019; Xia & Xie, 2019) or four categories (Rahhal et al., 2016; Sayantan et al., 2018) according to the AAMI standard; supraventricular ectopic beat (S), ventricular ectopic beat (V), fusion beat (F), unknown beat (Q), and any heartbeat not in S, V, F, and Q (N). The other is the detection of a single disease, He et al. (2021) detected myocardial infarction based on beats, whereas Shi et al. (2020) detected atrial fibrillation based on rhythms. The deep learning models used in this study were Convolutional Neural Networks (CNN) (Shi et al., 2020; Xia & Xie, 2019), Recurrent Neural Networks(RNN) (Wang et al., 2019), Deep Belief Networks(DBN) (Sayantan et al., 2018), and Deep Neural Networks(DNN) (Rahhal et al., 2016). For the query strategy in active learning, Wang et al. (2019) used information entropy and margin sampling, Rahhal et al. (2016) chose Entropy, and Breaking-Tie (BT) measurements, Sayantan et al. (2018) used BT measurement, Xia and Xie (2019) used modified BT, He et al. (2021) chose the Least Confidence, and Shi et al. (2020) used the Modified Entropy(MEN) measurement.

In summary, the current application of AL in ECG detection primarily improves the performance of the model, and the number of categories to be queried is small. Among them, detection based on heartbeat has at most 5-classification problems, whereas rhythmic-based detection is also a single disease detection. Most of the query strategies in the aforementioned studies are simple references or corrections to the uncertainty measurement and lack systematic comparison and analysis. When there are many categories of datasets to be queried, the uncertainty measurement has limitations, which may easily lead to local overfitting.

To reduce the annotation cost and develop a model with low dependence on labeled data, this study proposes a heartbeat detection algorithm based on deep active learning (DAL), that is, a CNN-based DAL model combining AL and DL. By analyzing classical query strategies, a series of stable and high-performance query strategies suitable for ECG are proposed, which are referred to as weak stratification query strategies based on morphological statistical features. The algorithm can reduce the amount of labeled data as much as possible while ensuring the high performance of the model, avoiding invalid and redundant annotations, and obtaining high-quality annotations. Simultaneously,

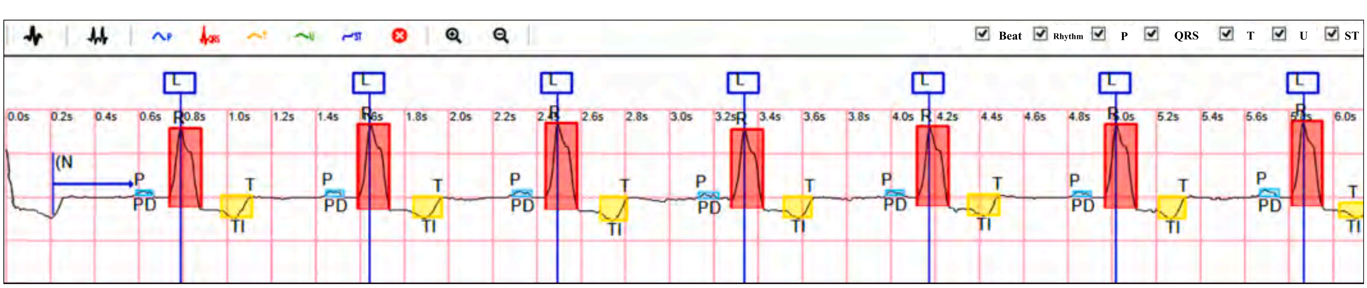


Fig. 1. The image is an annotation interface of the annotation system developed by this paper. The signal in the image is lead I, including beat, rhythm, and morphology labels. Beat annotation interpretation: L → Left bundle branch block beat. Rhythm annotation interpretation: (N → normal sinus rhythm. 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; TI → T wave inversion.

data with high interest and contribution to modeling were selected as the samples to be annotated, which is of great significance in reducing the annotation cost.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 presents a detailed description of the proposed algorithm. Sections 4 and 5 present the experimental settings and results, respectively. Finally, section 6 concludes the paper.

2. Related works

2.1. Deep Learning, active learning and deep active learning

Deep learning (DL) is a type of machine learning that originated from artificial neural networks and breaks the limitations of traditional neural networks on the number of layers, and the number of network layers can be selected according to the designer's needs. The motivation is to establish a model to simulate the neural connection structure of the human brain. When processing data such as images, voice, and text, the features are described through multiple transformation stages, and then the interpretation of the data is provided. Compared with the method of constructing features by artificial rules, using big data to learn features is more capable of characterizing the rich internal information of the data. And the performance of deep learning under massive data supply is much better than machine learning. The commonly used deep learning models include Recurrent Neural Networks(RNN), Deep Neural Networks(DNN), Deep Belief Networks(DBN), Convolutional Neural Networks (CNN), Generative Adversarial Networks(GAN), etc.

Active learning (AL) is a subfield of machine learning, which is a strategy or algorithm that can be interactively queried, also known as query learning or optimal experimental design(Settles, 2009). The model can actively select the data it wants to learn, select the most useful samples from unlabeled datasets, and hand them over to annotators for annotation, thereby minimizing annotation costs while maintaining performance.

Deep learning has a powerful learning ability, but it is highly greedy for labeled data, and its learning ability is completely determined by the quantity and quality of the labeled data. Active learning (Settles, 2009) obtains high-value labeled samples by actively selecting samples for annotation, which can effectively reduce annotation costs. Therefore, the deep active learning(DAL) model combined by deep learning(DL) and active learning(AL) is expected to reduce the demand for labeled data while maintaining a high learning performance.

Fig. 2 shows a comparison of the classical process of AL and DAL, where U represents the unlabeled dataset and L represents the labeled dataset (Ren et al., 2021).

Fig. 2(a) shows the AL process, which is directly querying samples in U according to the query strategy, and gives them to ECG experts for annotation. After annotation, the newly labeled data were added to L, the model was trained, while updating U and deleting the queried samples in U. This process is repeated until the U is empty or a pre-set termination condition is reached.

Fig. 2(b) shows the DAL process, the samples in U are first extracted features by the DL model and then queried based on a specific query strategy and sent to ECG experts for annotation. After annotation, the newly labeled data were added to L, the DL model was trained on L, while U is updated and the queried samples are deleted in U. After annotation, add the newly labeled data to L and train the DL model on L, while updating U and deleting the queried samples in U. This process is repeated until the unlabeled dataset is empty or a pre-set termination condition is reached.

In summary, the most significant difference between AL and DAL is that DAL first extracts features through DL and then selects samples based on a specific query strategy, thus making full use of DL's powerful data processing and feature extraction capabilities of DL.

2.2. Query strategies

The most widely used classical query strategies are Least Confidence (Wang & Shang, 2014), Margin Sampling (Wang & Shang, 2014), Entropy Sampling (Wang & Shang, 2014), K-Means Sampling, and K-Centers Greedy (Sener & Savarese, 2018).

(1) Least Confidence: Select the sample with the largest prediction probability but low reliability of the model, denoted as lc. The calculation method is shown in Eq. (1).

$$lc = \mathrm{argmax}_{i=1,\dots,n}(1 - P(\hat{y}|x_i))\ \hat{y} = \mathrm{argmax}_{j=1,\dots,m}P\left(y_j|x_i\right) \tag{1}$$

where x_i represents the i-th sample, y_j the j-th class, and $\hat{y}$ the largest posterior probability among the classes.

(2) Margin Sampling: Select the samples with the smallest probability difference between the largest and the next largest predicted by the model, denoted as *Margin*, and the calculation method is shown in Eq. (2).

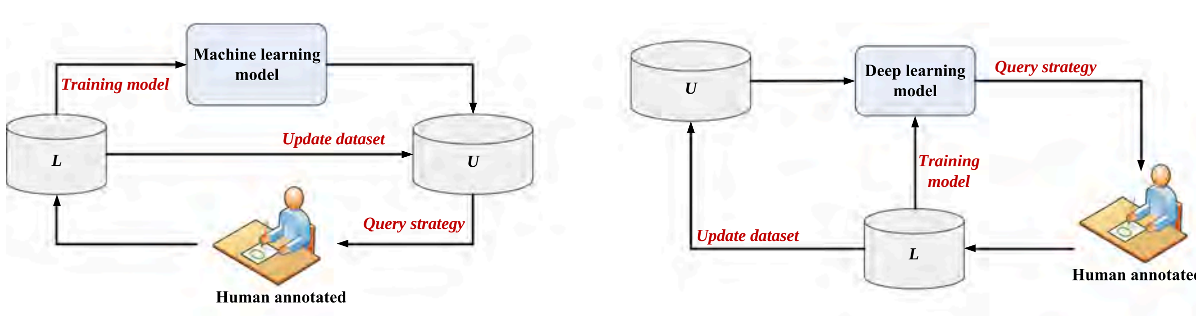
$$Margin = \mathrm{argmin}_{i=1,\dots,n}(P(\hat{y}_1|x_i) - P(\hat{y}_2|x_i)) \tag{2}$$

where x_i represents the i-th sample, $\hat{y}_1$ and $\hat{y}_2$ represent the two largest posterior probabilities across all the classes.

(3) Entropy Sampling: Entropy can be used to measure the uncertainty. A higher entropy indicates greater uncertainty, whereas a lower entropy indicates less uncertainty. Therefore, a sample with a higher entropy can be selected as the data to be annotated, denoted as *Entropy*. The calculation method is shown in Eq. (3).

$$Entropy = \mathrm{argmin}_{i=1,\dots,n} - \sum_j P\left(y_j|x_i\right) \tag{3}$$

where x_i represents the i-th sample, y_j represents the j-th class.



(a) Active learning (AL) process (b) Deep Active learning (DAL) process

Fig. 2. Comparison of typical AL and DAL models.

(4) K-Means Sampling: According to the K-Means clustering algorithm, the dataset is divided into k clusters, and the calculation method for the newly added sample u to be labeled is shown in the calculation Eq. (4).

$$u = \text{argmin} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{4}$$

where μ_i is the centroid of the cluster C_i.

(5) K-Centers Greedy: The strategy selects b samples in each round, finding the current best dataset by sequentially selecting b samples from the unlabeled dataset U and adding them to L. The newly added sample u must have the largest distance from L. The method for determining u is shown in Eq. (5).

$$u = \text{argmax}_{i \in [n] \setminus L} \text{min}_{j \in L} \Delta(x_i, x_j) \tag{5}$$

where x_i is the i-th sample in U, n is the total number of samples, x_j is the j-th sample in L. Newly added sample u must have the largest distance from L, and the method for determining the distance between u and L is to calculate the minimum distance between u and each sample of L.

3. Methodology

3.1. Overview of the algorithm

This study proposes a DAL beat detection algorithm based on the weak stratification query strategy for morphological statistical features. Fig. 3(Wang et al., 2021) shows the DAL-based ECG detection process, where U, L, and L_0 represent the unlabeled dataset, labeled dataset, and initially labeled dataset, respectively. The specific steps are as follows:

(1) To initialize the model, the CNN model was first pre-trained using the initially labeled training set L_0.
(2) Each iteration queries a specific number of samples from the unlabeled dataset U according to the weak stratification query strategy of morphological statistical features and is then annotated by ECG experts.
(3) After annotation, they were placed into labeled database L, the CNN model was trained and optimized again, while updating U and deleting the queried samples in U;
(4) Steps (2) and (3) are repeated continuously until the pre-set target is reached or the number of unlabeled data samples is zero.

In this study, the deep learning model used in the proposed algorithm is CNN. The structure of the CNN classifier includes four convolutional layers and two fully connected layers, as shown in Fig. 4, and the activation function used is ReLU.

3.2. Weak stratification query strategy based on morphological statistical features

The most obvious difference in the signals of various arrhythmias is the morphological difference, which is also the main basis for diagnosis. Therefore, this study proposes a query strategy based on weak stratification of morphological statistical features. Skewness is a digital feature of the asymmetry of a statistical data distribution. The skewness and direction of distribution were determined by measuring the skewness coefficient. Therefore, the skewness fully reflects the morphological characteristics of the signal. This study considers skewness as the main analysis object and proposes the following five query strategies based on skewness, setting the number of samples for each round of query as N and the number of arrhythmia types as C (which also represents the number of beat classes). To query samples in various distribution states as much as possible, five stratification strategies are proposed, including precise stratification, and strategy (3) “Skewhier-C” is divided into C layers; as well as general layering, strategy (1) “Skew” is divided into 1 layer, strategy (2) “Skewhierste” is divided into 2 layers, strategy (4) “Skewhier-C/2″ is divided into C/2 layers; also tried combining with other strategies like strategy (5) ”EntropySkw“ combined with information entropy. Detailed descriptions of the five strategies are as follows.

(1) Skew

The core idea of the “Skew” query strategy is to first calculate the skewness value of all the unlabeled samples $\{u_1, u_2, ..., u_m\}$, denoted as the set *Skew*, then sort the *Skew*. Finally, the first n samples were taken as samples to be labeled.

It is assumed that each sample contains t sampling points, denoted as $\{x_1, x_2, ..., x_t\}$, where x_i represents the i-th sampling point of the sample. The calculation method of the skewness value of the sample is shown in Eq. (6), and is denoted as *skew*:

$$skew = \frac{\sum_{i=1}^{t}(x_i - \mu)^3}{(t-1)\sigma^3} \quad \mu = \frac{x_1 + x_2 + ... + x_t}{t} \quad \sigma = \sqrt{\frac{\sum_{i=1}^{t}(x_i - \mu)^2}{t-1}} \tag{6}$$

Then the description of the “Skew” query strategy is shown in

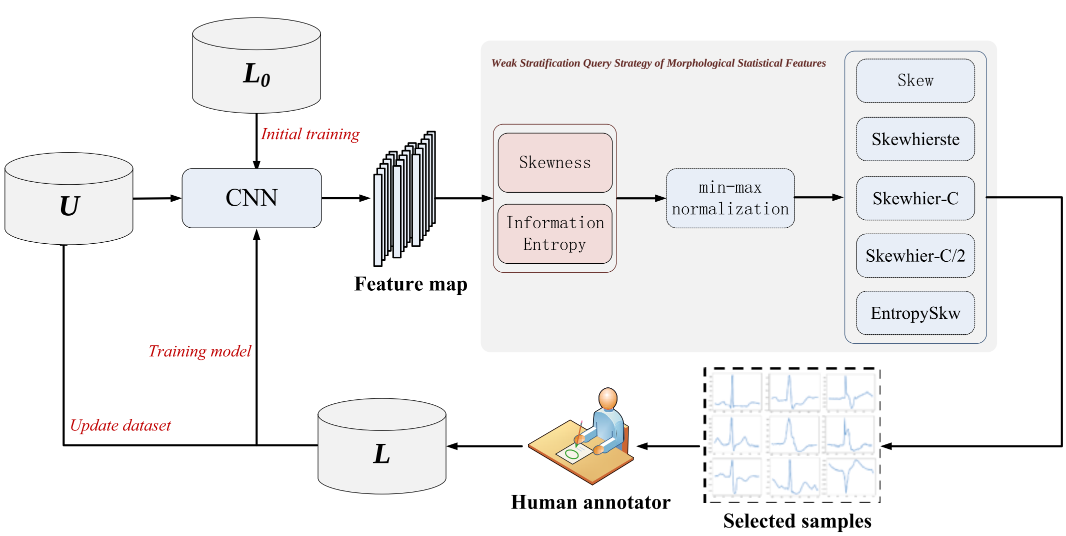


Fig. 3. The process of the DAL-based heartbeat detection algorithm proposed in this paper.

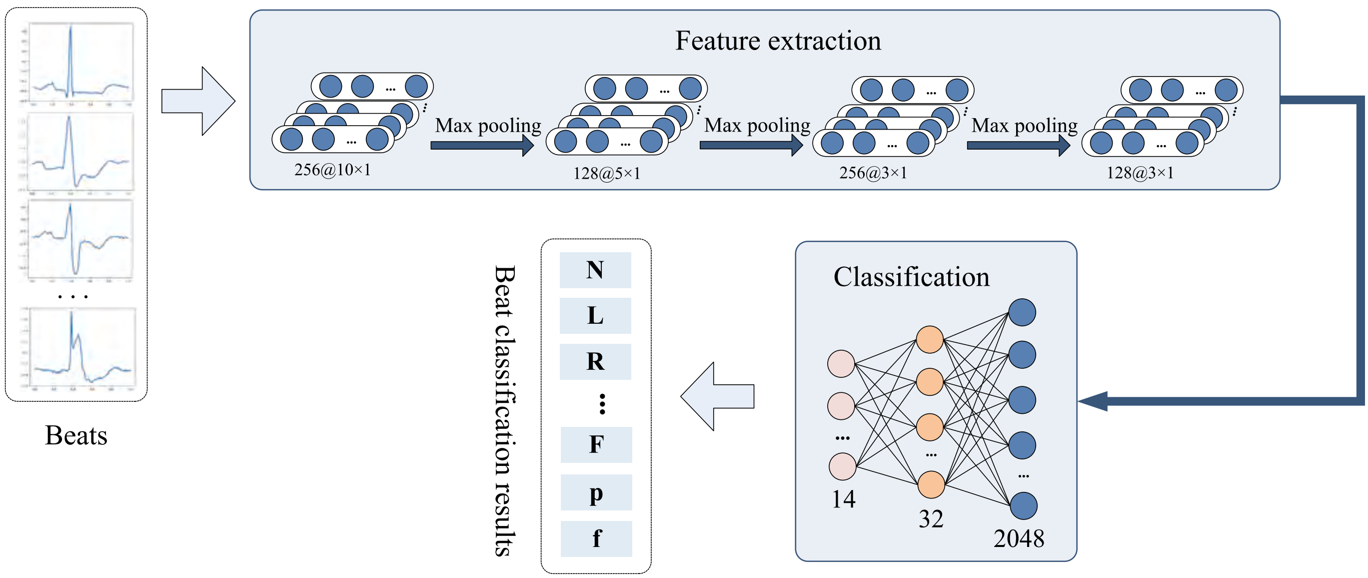


Fig. 4. The structure of CNN.

Algorithm 1.

Algorithm 1. "Skew" query strategy

Input: The set of containing m unlabeled samples $U = \{u_1, u_2, \ldots, u_m\}$.
Output: The set A containing n samples to be labeled.
1. Calculate the *skew* of all unlabeled samples in U according to formula (6), and obtain the set *Skew*:
   $Skew = \{skew_1, skew_2, \ldots, skew_m\}$
2. Normalized set *Skew*.
   Sort the set *Skew* and get the set *Oskew*: $Oskew = \mathrm{sort}(Skew)$;
3. $A = Oskew[0{:}n]$
4. Return A

(2) Skewhierste

The core idea of the "Skewierste" query strategy is to first calculate the *skew* of all unlabeled samples, sort them, and take $n/2$ samples at the head of the sequence and $n/2$ at the tail as the samples to be annotated. This process is presented in Algorithm 2.

Algorithm 2. "Skewhierste" query strategy

Input: The set of containing m unlabeled samples $U = \{u_1, u_2, \ldots, u_m\}$.
Output: The set A containing n samples to be labeled.
1. Calculate the *skew* of all unlabeled samples in U according to formula (6), and obtain the set *Skew*:
   $Skew = \{skew_1, skew_2, \ldots, skew_m\}$
2. Normalized set *Skew*.
3. Sort the set *Skew* and get the set *Oskew*:
   $Oskew = \mathrm{sort}(Skew)$;
3. For $j = 1, 2, 3, \ldots, n/2$
   $\mathrm{amin}_j = \min(Oskew)$; $\mathrm{A_{min}}$.append(amin_j); *Oskew*.remove(amin_j);
   $\mathrm{amax}_j = \max(Oskew)$; $\mathrm{A_{max}}$.append(amax_j); *Oskew*.remove(amax_j);
   EndFor
4. The set A is the sum of the minimum set and the maximum set: $A = \mathrm{Amin} + \mathrm{Amax}$;
5. Return A

(3) Skewhier-C

The strategy first calculates the *skew* of all the unlabeled samples and sorts them. C is the number of classes; the sequence is divided into C segments from beginning to end, each segment takes n/C samples, and n samples are taken as samples to be labeled. This process is presented in

Algorithm 3.

Algorithm 3. "Skewhier-C" query strategy

Input: The set of containing m unlabeled samples $U = \{u_1, u_2, \ldots, u_m\}$
Parameters: The number of classes: C.
Output: The set A containing n samples to be labeled.
Calculate the *skew* of all unlabeled samples in U according to formula (6), and obtain the set *Skew*:
$Skew = \{skew_1, skew_2, \ldots, skew_m\}$
Normalized set *Skew*.
Sort the set *Skew* and get the set *Oskew*:
$Oskew = \mathrm{sort}(Skew)$;
The number of unlabeled samples in each segment is $sn = m/C$, and the number of samples to be labeled in each segment is $ssn = n/C$;
For j in range(0, m, sn)
$B_j = Oskew[j{:}\ j + sn]$ $A_j = B_j[0{:}\ ssn]$ A.append(A_j)
EndFor
Return A

(4) Skewhier-C/2

The strategy first calculates the *skew* of all unlabeled samples and sorts them, then divides the sequence into $C/2$ segments from the head to the end, and takes $n/(C/2)$ samples from each segment, a total of n samples were taken as the samples to be annotated. This process is presented in Algorithm 3, where C is changed to $C/2$.

(5) EntropySkw

The strategy first calculates the *skew* and information entropy (Entropy) of all unlabeled samples, assigns weights w_1 and w_2, respectively, and calculates EntropySkw = *skew* * w_1 + Entropy * w_2. Sort EntropySkw and finally select the first n samples as the samples to be annotated.

The sample sequence was set to $\{x_1, x_2, \ldots, x_t\}$, and the symbol sets of the sequence, that is, the set of all values of the sampling point, were $\{q_1, q_2, \ldots, q_r\}$ and $t \geq r$. The information entropy of the sample, denoted as *entro*, is calculated as shown in Eq. (7).

$$entro = -\sum_{i=1}^{r} p_{(q_i)} \log_2\left(p_{(q_i)}\right) \tag{7}$$

The specific description of query strategy EntropySkw is shown in

Algorithm 4.

Algorithm 4. "EntropySkw" query strategy

Input: The set of containing m unlabeled samples $U = \{u_1, u_2, \ldots, u_m\}$.
Parameters: weights w_1, w_2.
Output: The set A containing n samples to be labeled.

1. Calculate the *skew* of all unlabeled samples in U according to formula (6), and obtain the set *Skew*:
   $Skew = \{skew_1, skew_2, \ldots, skew_m\}$
2. Normalized set *Skew*.
3. Calculate the *entro* of all unlabeled samples in U according to formula (7), and obtain the set *Entropy*:
   $Entropy = \{entro_1, entro_2, \ldots, entro_m\}$
4. Calculate the *entroskew*, and get the set *EntropySkew*:
   $entroskew_i = skew_i * w_1 + entro_i * w_2 \; I = (1,2,\ldots,m) EntropySkew = \{entroskew_1, entroskew_2, \ldots, entroskew_m\}$
5. Sort the set *EntropySkew* and get the set *SoEntropySkew*:
   $SoEntropySkew = \mathrm{sort}(EntropySkew)$
6. $A = SoEntropySkew[0{:}n]$
7. Return A

4. Experiment settings

4.1. Data collection

The database chosen for this study is the most widely used MIT-BIH arrhythmia database (Goldberger et al., 2000; Moody & Mark, 2001), collected from 47 subjects between 1975 and 1979. It contained 48 two-lead ECG recordings with a duration of half an hour and a sampling rate of 360 Hz. This database covers all the common cardiac arrhythmias and has become the gold standard for artificial intelligent ECG analysis.

To obtain a complete heartbeat, with a heart rate of 75 as the benchmark, the duration of each heartbeat is about 0.8 s. Therefore, taking the R wave peak as the center take 110 sampling points to the left and 175 sampling points to the right, then a heartbeat contains 286 sampling points. The specific heartbeat segmentation description is shown in Fig. 5. Using this method, 109,415 heartbeats were cut, thus covering 14 types of arrhythmias.

The beat distribution is shown in Table 1, wherein there are 75,019 "normal" beats, and 2 beats for "premature supraventricular or ectopic," it is obvious that the initial dataset is extremely unbalanced. To reduce the imbalance, and taking into account the largest number of abnormal heartbeats is "left bundle branch block," which is 8072. Therefore, it is determined that 8000 heartbeats are randomly selected from the "normal" heartbeats to participate in the experiment, and the rest of the categories use the number of real heartbeats. Ultimately, 42,396 beats, which cover 14 arrhythmia types, were obtained.

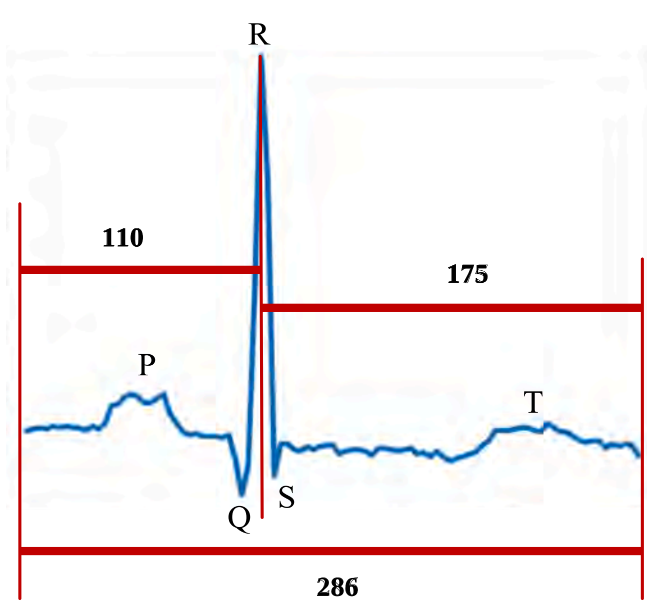


Fig. 5. Heart beat segmentation diagram.

Table 1
Description of beat distribution in MIT-BIH arrhythmia database.

Label	Description	Real beats	Used beats
N	Normal beat	75,019	8000
L	Left bundle branch block beat	8072	8072
R	Right bundle branch block beat	7255	7255
e	Atrial escape beat	16	16
j	Nodal (junctional) escape beat	229	229
A	Atrial premature beat	2546	2546
a	Aberrated atrial premature beat	150	150
J	Nodal (junctional) premature beat	83	83
S	Supraventricular premature or ectopic beat (atrial or nodal)	2	2
V	Premature ventricular contraction	7129	7129
E	Ventricular escape beat	106	106
F	Fusion of ventricular and normal beat	802	802
/	Paced beat	7024	7024
f	Fusion of paced and normal beat	982	982
Total		109,415	42,396

The test set was obtained from 42,396 heartbeats at a ratio of 33 %, and the remainder were used as the training set. The test set comprised 13,991 heartbeats. To facilitate iterative training, the training set was rounded off to 28,000 heartbeats. The experimental settings are listed in Table 2. First, 1000 heartbeats were extracted from the training set as the initial labeled dataset, and the rest were input into the unlabeled dataset. The model was first pre-trained using the initially labeled dataset. Each iteration queries 1000 heartbeats from the unlabeled dataset according to a specific query strategy and puts them into the labeled dataset with labels. Then, the labeled and unlabeled datasets were updated, and the model was retrained on the labeled dataset. Finally, the model is tested using the test set. Therefore, the classifier underwent 28 rounds of training and testing, as well as 27 rounds of querying, and the changes in model performance could be clearly observed as the sample size of the training set increased.

4.2. Normalization

Due to the fact that the proposed Skew series query strategies are all based on skewness, and the range of skewness values is $(-\infty,+\infty)$. Therefore, to generalize and unify the statistical distribution of the samples and limit the preprocessed data to a certain range for subsequent processing, it is necessary to normalize the skewness set $Skew = \{skew_1, skew_2, \ldots, skew_m\}$. The most commonly used normalization methods are Z-score normalization, Decimal Scaling normalization, and Min-max normalization, as shown in Eqs. (8)-(10), respectively, where $skew'_m$ represents the normalized data.

(1) Z-score normalization: Also known as standard deviation normalization, the processed data followed a standard normal distribution. However, it is difficult to map data to a fixed interval using this method. The calculation is as follows, where μ_{Skew} and σ_{Skew} are the average and standard deviation of the set *Skew*, respectively.

Table 2
Experiment Settings.

Training Set (Unlabeled dataset)				Test Set
Total	Initial labeled data set	Samples queried per round (manually annotated samples)	Total rounds	13,991
28,000	1000	1000	28	

$$skew'_m = \frac{skew_m - \mu_{Skew}}{\sigma_{Skew}} \tag{8}$$

(2) Decimal Scaling normalization: Move the decimal point to scale the data proportionally to fit within a specific range, calculated as follows, where j is the smallest integer that satisfies a maximum of $|skew'_m|$ less than 1.

$$skew'_m = \frac{skew_m}{10^j} \tag{9}$$

(3) Min-max normalization: This is a linear normalization method that maps data to the [0,1] interval and is calculated as follows:

$$skew'_m = \frac{skew_m - min_{Skew}}{max_{Skew} - min_{Skew}} \tag{10}$$

To compare the three normalization methods, the “Skew” query strategy was used, and the performance trends of classifiers K-nearest neighbor classifier (KNeighborsClassifier) and SVM with gaussian kernel (svm.SVC(kernel=’rbf’)) were compared and analyzed as the number of query rounds increased. To enhance the visualization effect, a 3D line chart was selected. The comparison results are shown in Fig. 6, it is obvious that the performance trends of the two classifiers are basically consistent, and the Min-max normalization is more stable. Min-max normalization also has the advantages of simple calculation, retaining the distribution characteristics of the original dataset, and mapping the data to fixed intervals for convenient subsequent processing. Therefore, in this study, Min-max normalization is chosen as the normalization method.

4.3. Model preparation

To verify the performance of the proposed DAL model and the applicability and stability of the proposed query strategies, we used classical machine learning algorithms for comparison and analysis. The model used was obtained from the Sklearn machine learning library (Pedregosa et al., 2011). Sklearn, also known as Scikit-learn, is a free machine learning library for the Python programming language. It contains various classical classification, regression, and clustering algorithms. The classifiers selected in this experiment include the decision tree classifier (DecisionTreeClassifier) and the regression tree classifier (DecisionTreeRegressor) in the decision tree category, as well as the random forest classifier (RandomForestClassifier), multilayer perceptron classifier (MLPClassifier), stochastic gradient descent classifier (SGDClassifier), SVM with linear kernel (svm.SVC(kernel=’linear’)), SVM with gaussian kernel (svm.SVC(kernel=’rbf’)), and K-nearest neighbor classifier (KNeighborsClassifier). The classifier used in the DAL model proposed in this study was based on CNN.

5. Experiment results

To comprehensively and deeply analyze the proposed DAL algorithm, three aspects were experimentally verified: (1) comparison and analysis of the proposed query strategies, (2) comparison and analysis of the proposed and classical strategies on various classifiers, and (3) performance comparison between the proposed CNN and other classical classifiers in active learning.

5.1. Distribution analysis of morphological statistical feature

The proposed Skew series query strategies are based on morphological statistical feature skewness, with the goal of querying samples in various distribution states. Therefore, the distribution of skewness directly determines the sample queried in each round under a specific querying strategy. From sections 3.2 and 4.2, it can be seen that after calculating the skewness of each sample, to facilitate subsequent processing, it is necessary to map the skewness to the interval [0,1] and record it as skew′. The training set contains 28,000 samples, firstly, 1000 samples are extracted as the initial labeled dataset, and the rest 27,000 are put into the unlabeled dataset. Therefore, the first round of querying requires calculating the skew′ of 27,000 samples. As shown in Fig. 7. and Table 3, the distribution histograms of Skew’ and corresponding analysis tables are presented. To make the analysis more intuitive, the interval with length of 0.028 is taken as the statistical unit, and the number of samples falling into the interval is counted. It is obvious that except for the 3 intervals close to “1″ with sample numbers of 20, 0, and 3, the skew′ values in all other intervals are above 200, with the highest reaching 1694. By dividing the interval [0,1] into three intervals with a spacing of approximately 0.336, we can obtain that the interval [0,0.336] contains 15,230 samples, (0.336, 0.672] contains 7730 samples, and (0.672, 1] contains 4040 samples. It can be concluded that the number of samples in the latter interval is about half of that in the previous interval.

Under such a distribution, which query strategy for extracting data can enable the model to achieve the highest stable performance fastest needs to be analyzed and verified by subsequent experiments.

5.2. Comparison and analysis of Skew series query strategies

The Skew series query strategies proposed in this study, are Skew, Skewierste, Skewier-C, Skewier-C/2 and EntropySkw. Because there were 14 types of heartbeats in the dataset, the value of C was 14. To obtain a comprehensive and systematic analysis, five strategies were applied to the deep learning CNN classifier and eight traditional machine learning classifiers. The performances obtained from each round of queries for the five strategies are shown in Fig. 8. It was set such that

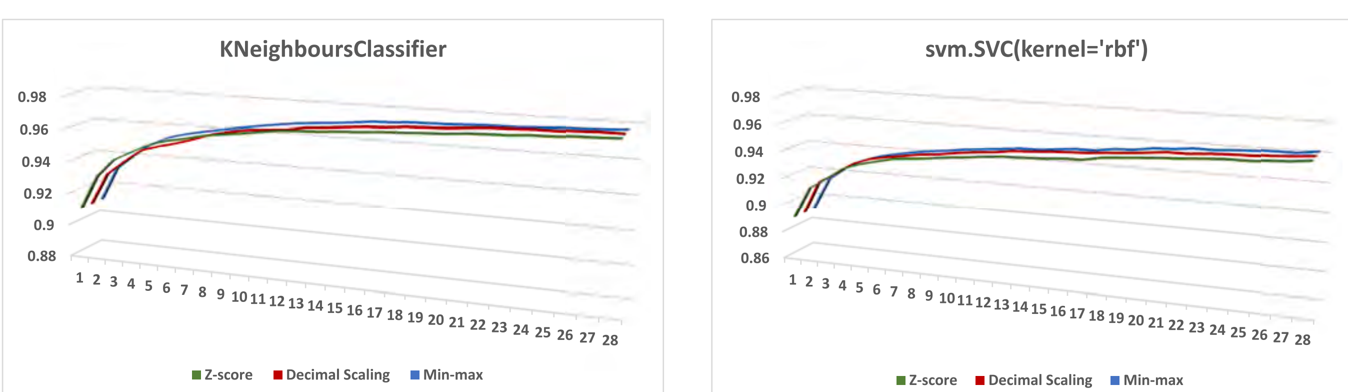


Fig. 6. Performance comparison of three normalization methods based on Skew query strategy on classifiers KNeighborsClassifier and SVM with gaussian kernel (svm.SVC(kernel=’rbf’)).

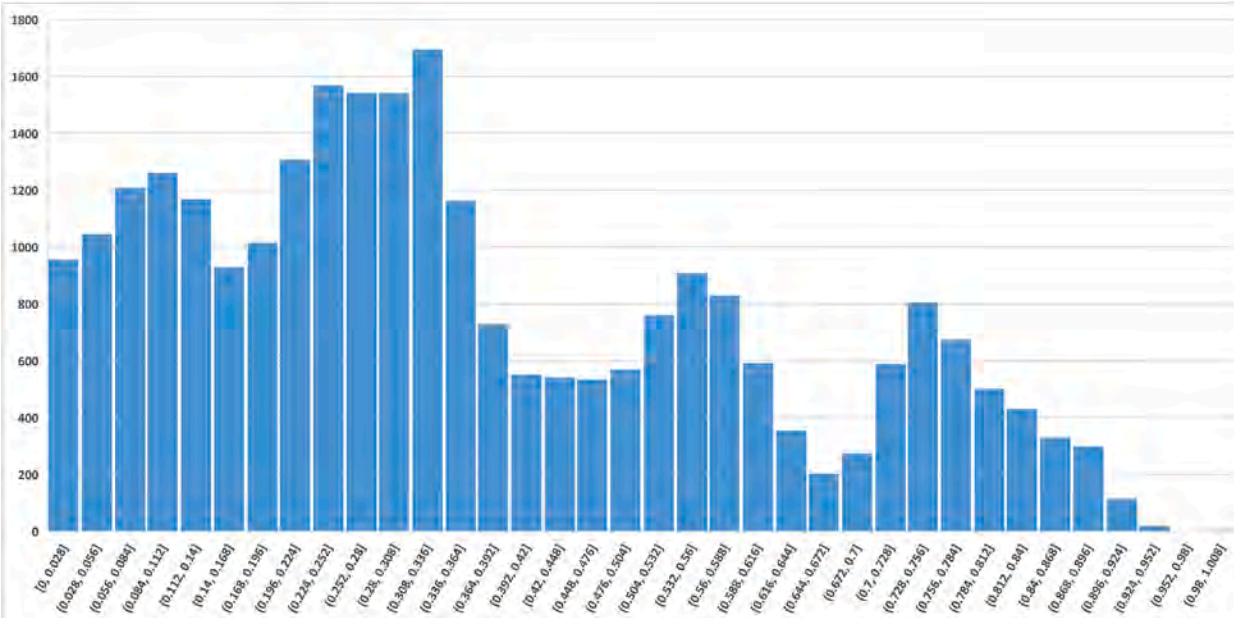


Fig. 7. Histogram of skew′distribution.

Table 3
Distribution analysis of morphological statistical feature skew′.

Large interval					
[0, 0.336]		(0.336, 0.672]		(0.672, 1]	
Short interval	No. of skew′	Short intervals	No. of skew′	Short interval	No. of skew′
[0, 0.028]	956	(0.336, 0.364]	1162	(0.672, 0.7]	274
(0.028, 0.056]	1045	(0.364, 0.392]	728	(0.7, 0.728]	588
(0.056, 0.084]	1208	(0.392, 0.42]	551	(0.728, 0.756]	805
(0.084, 0.112]	1260	(0.42, 0.448]	541	(0.756, 0.784]	674
(0.112, 0.14]	1168	(0.448, 0.476]	534	(0.784, 0.812]	502
(0.14, 0.168]	928	(0.476, 0.504]	568	(0.812, 0.84]	431
(0.168, 0.196]	1014	(0.504, 0.532]	760	(0.84, 0.868]	330
(0.196, 0.224]	1307	(0.532, 0.56]	908	(0.868, 0.896]	298
(0.224, 0.252]	1568	(0.56, 0.588]	829	(0.896, 0.924]	115
(0.252, 0.28]	1541	(0.588, 0.616]	591	(0.924, 0.952]	20
(0.28, 0.308]	1541	(0.616, 0.644]	355	(0.952, 0.98]	0
(0.308, 0.336]	1694	(0.644, 0.672]	203	(0.98, 1]	3
Total (27000)	15,230		7730		4040

there were 28,000 samples in the training set, the initial labeled samples were 1,000, and 1,000 samples were queried in each round. Therefore, the model underwent a total of 28 rounds of training, as shown on the abscissa. After each round of training, the same test set was used for testing, and the accuracy is shown on the vertical axis in Fig. 8.

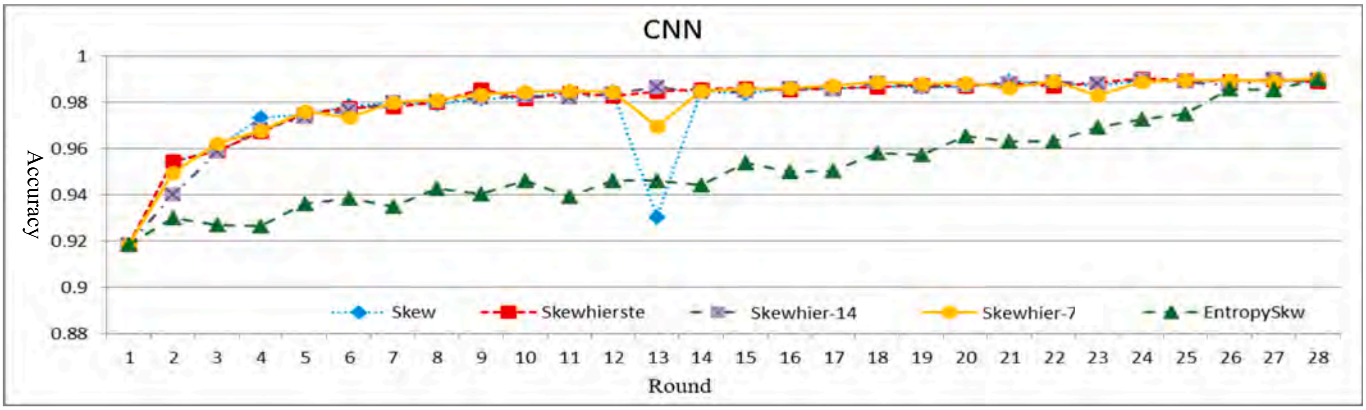
From Fig. 8, the stability of the Skew series query strategies and their impact on performance can be clearly observed. EntropySkew, which is combined with entropy, has the lowest performance, on most classifiers, it fluctuates significantly and cannot achieve the same high performance as that of the other strategies until the number of iterations reaches 26 or even 28. The strategies of Skewierste, Skewier-C (C = 14 in this experiment), and Skewier-C/2 were all relatively stable without major fluctuations, especially CNN, which almost reached the performance of the 28-th round in the 9-th round. For most classifiers, the performance of the Skew strategy is excellent; however, on the CNN, there is a large fluctuation in the 13-th iteration, which shows that its stability is slightly lower than that of Skewierste, Skewier-C and Skewier-C/2. Notably, in the SGDClassifier, all the strategies before the 14-th round of queries had different degrees of fluctuation, indicating that the model did not fully learn the knowledge before the 14-th round, and the high performance was accidental. In the 14-th round, the performance reached a steady state. Therefore, using the Skew query strategy on the SGDClassifier, only 50 % of the data achieved the performance obtained by training with all the data.

It can be seen that EntropySkew is not worth considering again, whereas Skew, Skewierste, Skewier-C, and Skewier-C/2 all perform well. In conclusion, based on the morphological statistical characteristics of Skew, taking one-way extreme values (Skew), taking the largest and smallest two-way extreme values (Skewhierste), precise stratification (Skewhier-C), and rough stratification (Skewhier-C/2) are all desirable solutions. Therefore, it is called a query strategy based on the weak stratification of morphological statistical features.

5.3. Comparison and analysis of Skew series and classical strategies

To analyze and verify the stability of the proposed strategy, two more representative strategies in the Skew series proposed in this paper, Skewierste and Skewier-C, were selected for comparison with the five most classical and widely used query strategies, namely, Margin Sampling(MarginSampling in the experiment), Least Confidence(LeastConfidence), Entropy Sampling(EntropySampling), K-Centers Greedy (KCenterGreedy), and K-Means Sampling(KMeansSampling). To obtain a more comprehensive verification and analysis, seven query strategies were combined with the CNN and eight classical classifiers, and the performances are shown in Fig. 9(a) and Fig. 9(b), respectively.

Fig. 9(a) and Fig. 9(b) show that EntropySampling is the worst,with lowest performance, excessive fluctuations on most classifiers and insufficient stability. Most classifiers do not achieve a performance as high as that of the other strategies until the number of iterations reaches 26 or 28. While LeastConfidemce and MarginSampling are better than



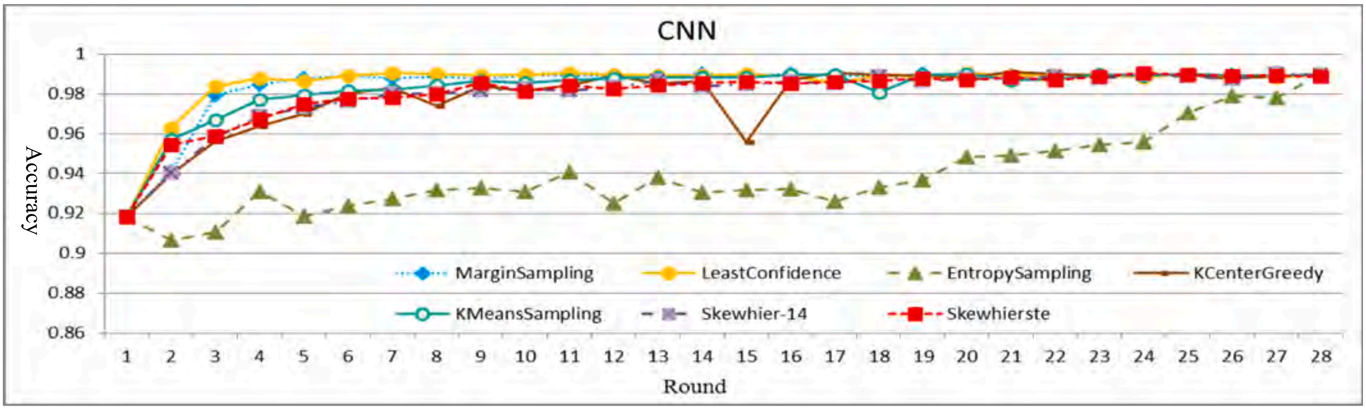
(a)

(b)

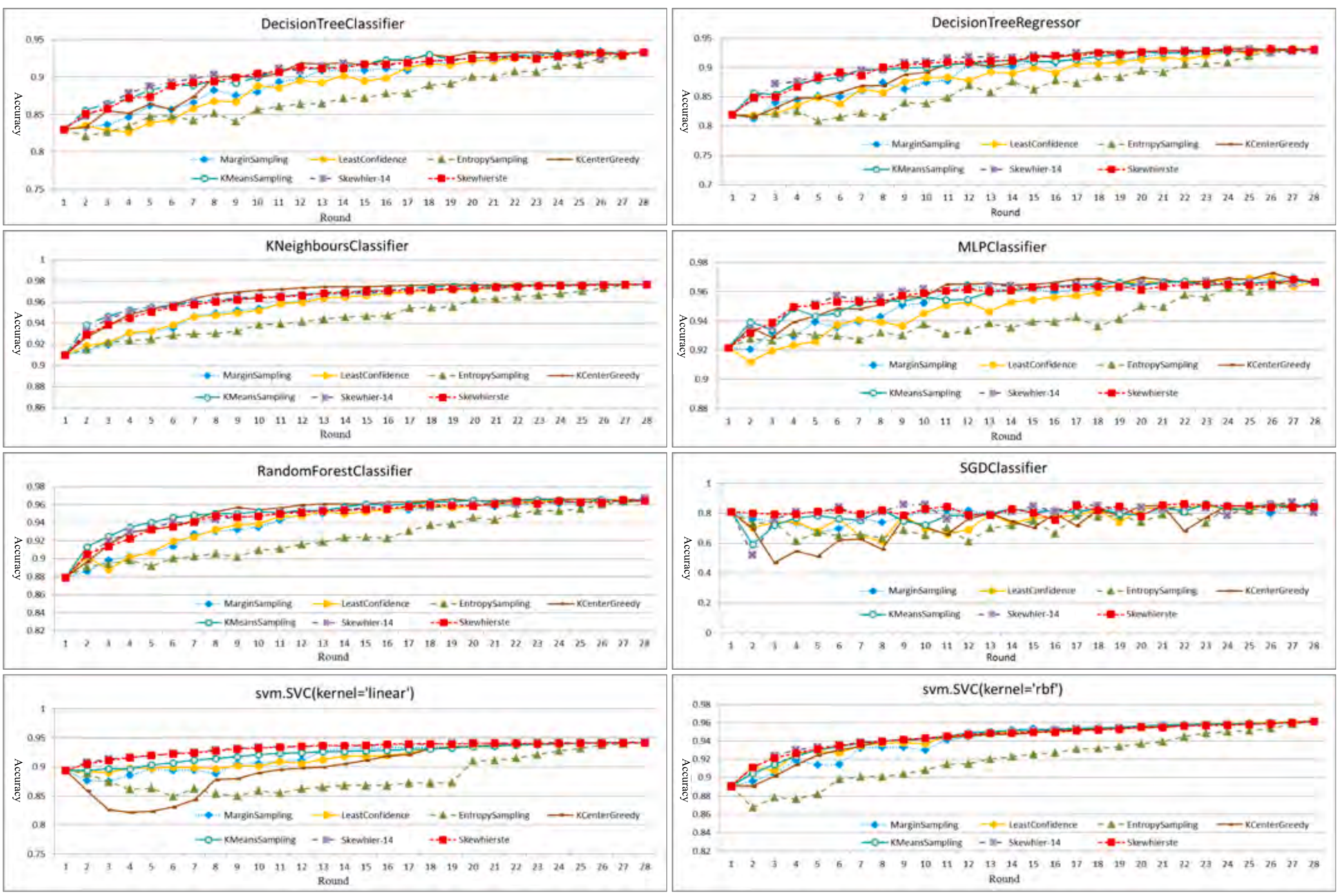
Fig. 8. The comparison between the proposed Skew series query strategies. (a) Line chart of the performance of Skew series query strategies on classifier CNN; (b) Line chart of the performance of Skew series query strategies on 8 classical classifiers.

EntropySampling, they are not sufficiently stable and not only fluctuate before reaching final stability, but most classifiers require more query rounds to reach the final stable state. Particularly, LeastConfidemce requires 18 rounds of query on KNeighborsClassifier, 22 rounds on DecisionTreeClassifier, 24 rounds on DecisionTreeRegressor, 18 rounds on svm.SVC(kernel='linear'), and 23 rounds on svm.SVC(kernel='rbf') to reach the highest stable state. The KCenterGreedy performs better than LeastConfidemce and MarginSampling, and achieves the best performance on the classifier KNeighborsClassifier, but is unstable and fluctuates on multiple classifiers, such as classifiers CNN, DecisionTreeClassifier, SGDClassifier, svm.SVC (kernel='linear') and the MLPClassifier, especially on svm.SVC(kernel='linear'), did not reach a steady state until the number of rounds reached 20. KMeansSampling is the best among several classical strategies; however, fluctuations still remain in SGDClassifier and MLPClassifier.

The proposed strategies, Skewierste and Skewier-C, are stable overall, except for the fluctuations in the SGDClassifier, which are more minor than those of the other strategies, indicating that the query performance of the proposed strategies is very stable. Notably, compared to other classical strategies, Skewierste and Skewier-C achieved the highest stable performance faster. Some classical strategies, such as KCenterGreedy, only perform well on one classifier, whereas the strategies proposed in this study can quickly reach the highest stable state for all the classifiers.



(a)



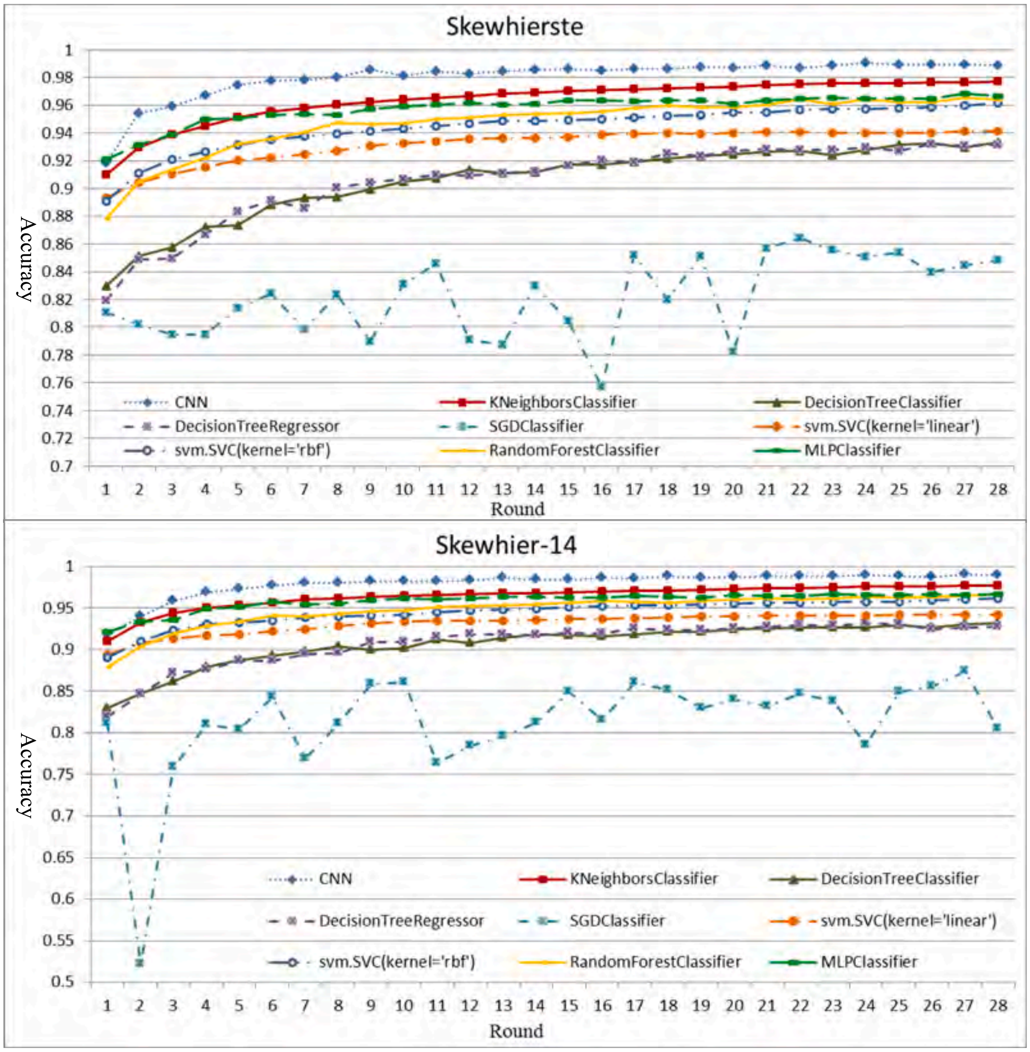
(b)

Fig. 9. Comparison of the proposed and the classical strategies. (a) Line chart of the performance of the proposed and the classical strategies on CNN. (b) Line chart of the performance of the proposed and the classical strategies on 8 classical classifiers.

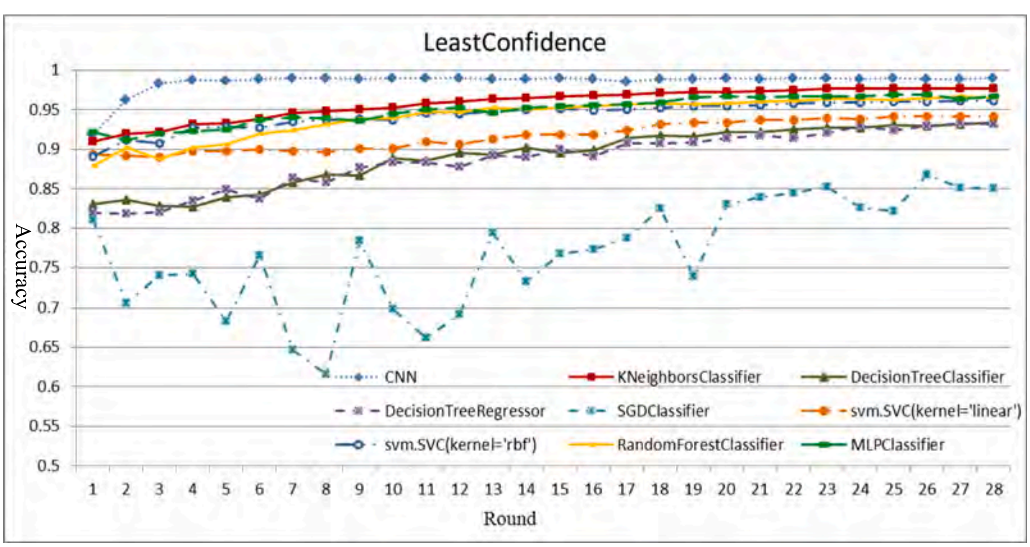
5.4. Performance comparison and analysis of CNN and classical classifiers

To verify the performance of the CNN classifier proposed in this paper, it is compared with 8 classical classifiers, which are DecisionTreeClassifier, DecisionTreeRegressor, RandomForestClassifier, MLPClassifier, SGDClassifier, svm.SVC (kernel = 'linear'), SVM.SVC (kernel = 'rbf') and NeighborsClassifier from the Sklearn machine learning library. At the same time, to make the comparison and analysis more comprehensive, the two most representative strategies Skewierste and Skewier-14, Additionally, the widely used classical strategy LeastConfidence with excellent performance are selected to analyze the performance and change trend obtained by each round of query on nine classifiers to verify the performance of the classifier CNN proposed in this paper. Fig. 10 shows the line chart of the performance analysis of the CNN and eight classical machine learning classifiers: Fig. 10(a) shows the line chart of the performance comparison between the CNN and eight classical classifiers on the Skewierste and Skewier-C strategies, and Fig. 10(b) shows the line chart that compares the performance of the CNN and eight classical classifiers on the LeastConfidence strategy.

To make the analysis more comprehensive and detailed, the accuracy of the LeastConfidence strategy in each round of query on the CNN and eight classical classifiers is liste in Table 4. In the table, KNeighborsClassifier is abbreviated as KNC, DecisionTreeClassifier is abbreviated



(a)



(b)

Fig. 10. Performance comparison of CNN and 8 classical machine learning classifiers. (a) Line graph of the performance of CNN and 8 classical classifiers on the proposed strategies Skewierste and Skewier-C; (b) Line graph of the performance of CNN and 8 classical classifiers on the strategy LeastConfidence.

Table 4
The accuracy of CNN and 8 classical classifiers on the strategy LeastConfidence per query round (%).

Round	CNN	KNC	DTC	DTR	SGD	SVML	SVMR	RFC	MLP
1	91.9	91.0	83.0	82.0	81.1	89.3	89.1	87.9	92.1
2	96.3	91.9	83.6	81.9	70.7	89.2	91.1	90.2	91.2
3	98.4	92.2	82.8	82.1	74.0	89.0	90.7	88.8	92.0
4	98.8	93.1	82.7	83.5	74.3	89.7	92.7	90.2	92.3
5	98.7	93.3	83.9	84.9	68.3	89.8	92.9	90.7	92.6
6	98.9	93.9	84.3	83.8	76.6	89.9	92.7	91.9	93.7
7	99.0	94.6	85.8	86.3	64.7	89.8	93.5	92.5	94.1
8	99.0	94.8	86.8	85.8	61.6	89.7	93.7	93.2	93.9
9	98.9	95.0	86.7	87.6	78.4	90.1	93.8	93.8	93.7
10	99.0	95.2	88.9	88.3	69.9	90.1	93.7	93.9	94.5
11	99.0	95.9	88.5	88.4	66.2	90.9	94.6	94.7	95.1
12	99.0	96.0	89.6	87.8	69.1	90.6	94.5	94.7	95.3
13	98.9	96.4	89.3	89.2	79.4	91.3	94.8	95.2	94.6
14	98.9	96.5	90.2	89.0	73.3	91.8	95.0	95.0	95.3
15	99.0	96.7	89.5	89.9	76.8	91.8	95.2	95.2	95.4
16	98.9	96.9	89.9	89.1	77.3	91.9	94.9	95.4	95.6
17	98.6	96.9	91.4	90.7	78.8	92.4	95.1	95.8	95.7
18	98.9	97.1	91.7	90.7	82.5	93.2	95.3	95.8	95.9
19	98.9	97.2	91.6	90.9	73.9	93.4	95.5	95.7	96.6
20	99.0	97.2	92.2	91.4	83.1	93.4	95.5	95.8	96.7
21	98.9	97.3	92.2	91.7	83.9	93.7	95.6	96.1	96.6
22	99.0	97.5	92.5	91.4	84.4	93.7	95.8	96.1	96.7
23	99.0	97.7	92.7	92.1	85.3	93.9	95.9	96.4	96.7
24	98.8	97.7	92.7	92.9	82.6	93.9	95.9	96.4	96.7
25	99.0	97.6	93.1	92.4	82.2	94.2	96.0	96.3	96.9
26	98.9	97.7	92.9	93.0	86.8	94.2	96.0	96.4	96.9
27	98.8	97.7	93.2	93.3	85.1	94.2	96.1	96.6	96.4
28	99.0	97.7	93.3	93.1	85.0	94.1	96.2	96.4	96.7

as DTC, DecisionTreeRegressor is abbreviated as DTR, SGDClassifier is abbreviated as SGD, svm.SVC(kernel='linear') is abbreviated as SVML, svm.SVC(kernel='rbf') is abbreviated as SVMR, RandomForestClassifier is abbreviated as RFC, and MLPClassifier is abbreviated as MLP.

As shown in Fig. 10, among three query strategies, the accuracy of the classifier CNN in each round is higher than that of the other eight classical classifiers, and the overall performance is also significantly higher than other classifiers. This indicates that CNN has advantages over classical classifiers in terms of performance. At the same time, it can be observed that the performance of the three strategies is greatly improved in the first few queries, and then tends to be stable with a small increase, which is more obvious for the Skewierste and Skewier-14 strategies. Fig. 10(a) is smoother than Fig. 10(b), again indicating that Skewierste and Skewier-14 are more stable than LeastConfidence. Particularly, LeastConfidence performed very well on the CNN, but not as well as Skewierste and Skewier-14 on other classifiers.

Notably, as seen from Table 4, the accuracy of the strategy LeastConfidence on the classifier CNN reached 98.8 % in the 4-th round, and the subsequent iterations are only slightly improved on the basis of it. In other words, the CNN can use only 4,000 samples to achieve the performance obtained by training the model with 28,000 samples, thus saving (28000–4000)/28000*100 %=85.7 % of the labeled samples and annotation workload. Additionally, it can be observed that the that DAL plays a pivotal role in reducing annotation costs. For the Skewierste and Skewier-14 strategies, the accuracy reached 98.0 % in the 7-th or 8-th round, and the subsequent iterations were also slightly improved. To justify, it is determined that the 9-th iteration reached a stable convergence state, so that 67.9 % of the labeled samples and annotation costs can be saved. In summary, it is of great practical and scientific value to query out the samples with the highest interest and greatest contribution to the model improvement as the samples to be annotated.

5.5. Discussion

According to the above experimental analysis, compared with the classical strategies Margin Sampling, Least Confidence, Entropy Sampling, K-Centers Greedy, and K-Means Sampling, the query strategies Skew, Skewierste, Skewier-C and Skewhier-C/2 proposed in this paper based on weak stratification of morphological statistical features, have better overall stability and less fluctuation and achieves the highest stable state as a faster rate. Moreover, compared with the classical strategy, the Skew series strategy proposed in this paper has better adaptability to a variety of classifiers.

Active learning has been applied to artificial intelligence detection of ECG. Table 5 shows the analysis and comparison of previous methods since 2010 and the method proposed in this study. For a more comprehensive analysis, the detection type, query strategies, classifiers used in the active learning, specific numbers of classes, and performance analysis are listed in detail for each method. The query strategies of the listed previous methods were mostly the same as those in the comparative experiments in this study, where EN denotes Entropy, LC denotes Least Confidence, MS denotes Margin Sampling. K-CG is the abbreviation for K-Centers Greedy and K-MS is short for K-Means Sampling. As for other strategies, MEN represents Modified Entropy, HAC is the abbreviation for Hierarchical Clustering, BT (Breaking-Ties) is the same as PPS (Posterior Probability Sampling), both are based on posterior probability sampling, MBT represents modified BT, and QBC is a query based on committee votes. The classifiers are all deep learning classifiers, except Pasolli and Melgani (2010) and Wiens and Guttag (2010), who chose SVM, where DBN stands for deep belief network and BiLSTM stands for bidirectional long short-term memory. In performance analysis, the research focus is also different: some analyze how much annotation cost is reduced, some analyze how much contribution to performance improvement is made, and others pursue how much overall performance is achieved. Among them, SVEB represents "S"(supraventricular) class to all other classes, and VEB represents "V"(ventricular) class to all other classes, which are essentially binary classifications.

From Table 5, it can be observed that the application targets of active learning in artificial intelligence detection of ECG can be divided into the following three categories. Firstly, reduce the annotation cost, such as Pasolli and Melgani (2010), Wiens and Guttag (2010) and this paper; The second is to improve the detection performance of specific tasks, such as Rahhal et al. (2016), Sayantan et al. (2018), Wang et al. (2019) and Xia and Xie (2019); The third is to improve the overall detection

Table 5
Analysis and comparison of the proposed and previous methods for ECG detection based on active learning on the MIT-BIH Arrhythmia Database.

Method	Year	Detection type	Query strategy	classifier	The num. of classes	Performation analysis
(Pasolli & Melgani, 2010)	2010	Heart beat	MS,PPS,QBC	SVM	6	Saving 88.3 % annotation cost
(Wiens & Guttag, 2010)	2010	Heart beat	HAC	SVM	2	Saving over 90 % annotation cost
(Rahhal et al., 2016)	2016	Heart beat	EN, BT	DNN	2	SVEB accuracy improved by 4.9 %; VEB accuracy improved by 2.1 %
(Sayantan et al., 2018)	2018	Heart beat	BT	DBN	2	SVEB accuracy improved by 2.05 %; VEB accuracy improved by 1.32 %
(Wang et al., 2019)	2019	Heart beat	EN + MS	RNN	2	SVEB accuracy improved by 18.2 %; VEB accuracy improved by6.2 %
(Xia & Xie, 2019)	2019	Heart beat	MBT	CNN	4	Overall accuracy 99.21 % SVEB accuracy improved by 1.81 %; VEB accuracy improved by 1.21 %
(Jin et al., 2021)	2021	Heart beat	RBT	CNN+ BiLSTM	4	Overall accuracy 99.34 % Accuracy improved by 13.23 %
This paper	2022	Heart beat	Skew series, MS, LC, EN,K-CG, K-MS	CNN	14	Overall accuracy **99 %** Saving **85.7 %** annotation cost

performance of the model, such as Xia and Xie (2019), Jin et al. (2021) and this paper. Table 5 details the querying strategies and classifiers used by each study, as well as analyzes and compares the performance of each study under different class numbers. The number of classifications in this study was 14, which is the highest, and other studies ranged from binary classification to 6-classification. The highest annotation cost saved in this study was 85.7 %, although lower than that of Pasolli and Melgani (2010) and Wiens and Guttag (2010), and the labeled sample size required for 14-classification must be higher than that for binary classification and 6-classification. This is because 14-classification requires 14 types of labeled samples, whereas binary classification requires only two types of labeled samples. Obviously, the method proposed in this paper is highly advantageous in terms of saving annotation costs and reducing the annotation workload. Although the overall accuracy was not the highest, 99 % of the 14-classification were very competitive compared to 99.21 % of the 4-classification by Xia and Xie (2019) and 99.34 % of the 4-classification by Jin et al. (2021).

6. Conclusions and future directions

The annotation cost of ECG data is much higher than that of ordinary target recognition, which leads to a shortage of ECG label data in terms of both quantity and type, thus significantly limiting the construction of excellent models. Reducing the annotation cost and model demand for labeled data is an urgent problem to be solved. Therefore, this study introduces active learning and proposes a heartbeat detection algorithm based on DAL. Based on the characteristics of ECG signals, the Skew series of query strategies based on weak stratification of statistical features suitable for ECG was proposed, namely, Skew, Skewierste, Skewier-C and Skewier-C/2. Simultaneously, to ensure the desired performance, a CNN-based classifier was constructed. The experiments verified that the query strategy proposed in this study has higher stability and adaptability compared to other classical strategies and that the performance of the proposed CNN is also higher than that of other classical classifiers. Compared to previous research, it has a higher competitive advantage in terms of the number of classifications and cost savings of annotation.

With the increasing demand for daily ECG monitoring applications, computer-aided wearable ECG diagnostic technology has become a research focus. Due to the fact that in the field of daily ECG monitoring, computer-assisted detection targets long-term ECG segments, research cannot be limited to heart beats. Rhythm annotation and computer-assisted rhythm diagnostic techniques should also be considered.

In the future, it is necessary to continue to research rhythm detection algorithms based on DAL and further verify whether the Skew series strategies proposed in this paper are suitable for rhythm detection. For rhythms, we will focus on intelligent annotations and diagnostic algorithms. Based on this study, we optimize the proposed Skew series query strategies or propose new query strategies suitable for rhythm, and research better DAL models to reduce the cost of rhythm annotation and achieve high-performance rhythm detection. In clinical applications, one ECG segment may contain multiple arrhythmias, multi-label rhythm detection should also be the focus of future research.

CRediT authorship contribution statement

Haiyan Wang: Conceptualization, Methodology, Validation, Writing – original draft. **Yanjie Zhou:** Conceptualization, Formal analysis, Writing – review & editing. **Xiangdong Niu:** Data curation, Investigation. **Daijun Liu:** Visualization, Investigation, Resources. **Lingling Li:** Formal analysis, Investigation, Project administration, Funding acquisition. **Ying Duan:** Data curation, Validation, Investigation. **Zongmin Wang:** Conceptualization, Supervision.

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.

Acknowledgements

This work was supported by the by the Supported by the National Natural Science Foundation of China [grant number 62302463]; the Scientific Research Team Plan of Zhengzhou University of Aeronautics [grant number 23ZHTD01001]; the National Natural Science Foundation of China [grant number 62202434]; the Science and Technology Project of Henan Province in China [grant number 222102210079]; the Collaborative Education Project of Industry University Cooperation of the Ministry of Education of China [grant number 220506008202559]; the Collaborative Innovation Center of Henan Province (Aerospace electronic Information Technology).

References

Acharya, U. R., Fujita, H., & Oh, S. L. (2017). Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Information Sciences, 415*, 190–198.
Acharya, U. R., Oh, S. L., & Hagiwara, Y. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine, 89*, 389–396.
Andersen, R. S., Peimankar, A., & Puthusserypady, S. (2019). A deep learning approach for real-time detection of atrial fibrillation. *Expert Systems With Applications, 115*, 465–473.
ANSI/AAMI (2008). Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms, Association for the Advancement of Medical Instrumentation (AAMI), *American National Standards Institute*, Inc. (ANSI), ANSI/AAMI/ISO EC57, 1998-(R).
Asgari, S., Mehrnia, A., & Moussavi, M. (2015). Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. *Computers in Biology and Medicine, 60*, 132–142.

Attia, Z. I., Noseworthy, P. A., & Lopez-Jimenez, F. (2019). An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: A retrospective analysis of outcome prediction. *The Lancet, 394* (10201), 861–867.

Berkaya, S. K., Uysal, A. K., & Gunal, E. S. (2018). A survey on ECG analysis. *Biomedical Signal Processing and Control, 43*, 216–235.

Bozkurt, F., Uçar, M. K., & Bozkurt, M. R. (2020). Detection of abnormal respiratory events with single channel ECG and hybrid machine learning model in patients with obstructive sleep apnea. *Irbm, 41*(5), 241–251.

Chazal, P. D., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering, 51*(7), 1196–1206.

Chen, L., Zhang, X., & Song, C. (2014). An automatic screening approach for obstructive sleep apnea diagnosis based on single-lead electrocardiogram. *IEEE Transactions on Automation Science & Engineering, 12*(1), 106–115.

Erdenebayar, U., Kim, Y. J., & Park, J. U. (2019). Deep learning approaches for automatic detection of sleep apnea events from an electrocardiogram. *Computer Methods and Programs in Biomedicine, 180*, Article 105001.

Ge, Z., Jiang, X., & Tong, Z. (2021). Multi-label correlation guided feature fusion network for abnormal ECG diagnosis. *Knowledge-Based Systems, 233*, Article 107508.

Geweid, G. G., & Chen, J. D. (2022). Automatic classification of atrial fibrillation from short single-lead ECG recordings using a Hybrid Approach of Dual Support Vector Machine. *Expert Systems with Applications, 198*, Article 116848.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation, 101*(23), e215–e220.

He, Z., Yuan, Z., An, P., Zhao, J., & Du, B. (2021). MFB-LANN: A lightweight and updatable myocardial infarction diagnosis system based on convolutional neural networks and active learning. *Computer Methods and Programs in Biomedicine, 210*, Article 106379.

Joshi, A.J., Porikli, F., Papanikolopoulos, N. (2009). Multi-class active learning for image classification. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp.2372–2379). IEEE.

Jovic, A., & Bogunovic, N. (2011). Electrocardiogram analysis using a combination of statistical, geometric, and nonlinear heart rate variability features. *Artificial Intelligence in Medicine, 51*(3), 175–186.

Jin, Y., Qin, C., Liu, J., Li, Z., Shi, H., Lin, K., Liu, Y., & Liu, C. (2021). A novel incremental and interactive method for actual heartbeat classification with limited additional labeled samples. *IEEE Transactions on Instrumentation and Measurement, 70*, 1–12.

Kim, J., Min, S. D., & Lee, M. (2011). An arrhythmia classification algorithm using a dedicated wavelet adapted to different subjects. *Biomedical Engineering Online, 10*(1), 1–19.

Kiranyaz, S., Ince, T., & Gabbouj, M. (2015). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering, 63*(3), 664–675.

Li, C., Bao, Z., Li, L., & Zhao, Z. (2020). Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multi-modal emotion recognition. *Information Processing & Management, 57*(3), Article 102185.

Li, Y. J., Tang, X. Y., Wang, A., & Tang, H. (2017). Probability density distribution of delta RR intervals: A novel method for the detection of atrial fibrillation. *Australasian Physical & Engineering Sciences in Medicine, 40*(3), 707–716.

Liu, X., Wang, H., & Li, Z. (2021). Deep learning in ECG diagnosis: A review. *Knowledge-Based Systems, 227*, Article 107187.

Llamedo, M., & Martínez, J. P. (2010). Heartbeat classification using feature selection driven by database generalization criteria. *IEEE Transactions on Biomedical Engineering, 58*(3), 616–625.

Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine, 20*(3), 45–50.

Pasolli, E., & Melgani, F. (2010). Active learning methods for electrocardiographic signal classification. *IEEE Transactions on Information Technology in Biomedicine, 14*(6), 1405–1416.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pławiak, P. (2018). Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system. *Expert Systems With Applications, 92*, 334–349.

Rahhal, M. M. A., Bazi, Y., & AlHichri, H. (2016). Deep learning approach for active classification of electrocardiogram signals. *Information Sciences, 345*, 340–354.

Raj, S., Ray, K. C., & Shankar, O. (2016). Cardiac arrhythmia beat classification using DOST and PSO tuned SVM. *Computer Methods and Programs in Biomedicine, 136*, 163–177.

Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2021). A survey of deep active learning. *ACM Computing Surveys (CSUR), 54*(9), 1–40.

Ribeiro, A. H., Ribeiro, M. H., & Paixão, G. M. M. (2020). Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications, 11*(1), 1–9.

Saadatnejad, S., Oveisi, M., & Hashemi, M. (2019). LSTM-based ECG classification for continuous monitoring on personal wearable devices. *IEEE Journal of Biomedical and Health Informatics, 24*(2), 515–523.

Sayantan, G., Kien, P. T., & Kadambari, K. V. (2018). Classification of ECG beats using deep belief network and active learning. *Medical & Biological Engineering & Computing, 56*(10), 1887–1898.

Sener, O., Savarese, S.(2018). Active Learning for Convolutional Neural Networks: A Core-Set Approach. *International Conference on Learning Representations (ICLR)*.

Settles, B.(2009). Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*.

Shi, H., Wang, H., & Qin, C. (2020). An incremental learning system for atrial fibrillation detection based on transfer learning and active learning. *Computer Methods and Programs in Biomedicine, 187*, Article 105219.

Taji, B., Chan, A. D. C., & Shirmohammadi, S. (2017). False alarm reduction in atrial fibrillation detection using deep belief networks. *IEEE Transactions on Instrumentation and Measurement, 67*(5), 1124–1131.

Tan, J. H., Hagiwara, Y., Pang, W., Lim, I., Oh, S. L., Adam, M., Tan, R. S., Chen, M., & Acharya, U. R. (2018). Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals. *Computers in Biology and Medicine, 94*, 19–26.

Thakor, N. V., Zhu, Y. S., & Pan, K. Y. (1990). Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm. *IEEE transactions on biomedical engineering, 37*(9), 837–843.

Wang, D., Shang, Y.(2014). A new active labeling method for deep learning. *2014 International Joint Conference on Neural Networks (IJCNN)*(pp. 112-119).

Wang, G., Zhang, C., & Liu, Y. (2019). A global and updatable ECG beat classification system based on recurrent neural networks and active learning. *Information Sciences, 501*, 523–542.

Wang, H., Zhou, Y., Zhou, B., Wang, Z.M.(2021). A Novel Method for Detection of ECG with Deep Learning. *2021 7th International Conference on Computer and Communications (ICCC)*(pp. 631-635).

Wang, H. Y., Dai, H. H., Zhou, Y. J., Zhou, B., Lu, P., Zhang, H. P., & Wang, Z. M. (2021). An effective feature extraction method based on GDS for atrial fibrillation detection. *Journal of Biomedical Informatics, 103819*.

Wang, H. Y., Zhou, Y. J., Zhou, B., Niu, X. D., & Wang, Z. M. (2021). Interactive ECG annotation: An artificial intelligence method for smart ECG manipulation. *Information Sciences, 581*, 42–59.

Wang, N., Feng, P., Ge, Z., Zhou, Y., Zhou, B., & Wang, Z. (2023). Adversarial Spatiotemporal Contrastive Learning for Electrocardiogram Signals. *IEEE Transactions on Neural Networks and Learning*. Systems,1–15.

Wiens, J., & Guttag, J. (2010). Active learning applied to patient-adaptive heartbeat classification. *Advances in neural information processing systems, 23*, 1–9.

Xia, Y., & Xie, Y. (2019). A novel wearable electrocardiogram classification system using convolutional neural networks and active learning. *IEEE Access, 7*, 7989–8001.

Yao, Q., Wang, R., & Fan, X. (2020). Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network. *Information Fusion, 53*, 174–182.

Yin, J., Xiao, P., Li, J., Liu, Y., Yan, C., & Zhang, Y. (2020). Parameters analysis of sample entropy, permutation entropy and permutation ratio entropy for RR interval time series. *Information Processing & Management, 57*(5), Article 102283.

Zhai, X., Zhou, Z., & Tin, C. (2020). Semi-supervised learning for ECG classification without patient-specific labeled data. *Expert Systems with Applications, 158*, Article 113411.

Zhou, X., Ding, H., Ung, B., Pickwell-MacPherson, E., & Zhang, Y. (2014). Automatic online detection of atrial fibrillation based on symbolic dynamics and Shannon entropy. *Biomedical Engineering Online, 13*(1), 1–18.