Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 20, No. 3, 2026

EC Classification Algorithm Based on Deep Ensemble Learning

Jingli Gao

School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, China

Abstract: Cardiovascular disease has become a major global public health concern due to its high mortality rate. As a core non-invasive diagnostic tool, electrocardiogram (ECG) automatic classification is critical for the early screening of heart diseases. To address the problems of low efficiency in manual diagnosis, the inability of single deep learning models to simultaneously capture morphological and temporal features in ECG classification, and insufficient robustness under class-imbalanced scenarios, this paper proposes a deep ensemble learning algorithm named CBR-Stacking. The algorithm uses 1D-CNN, CNN-BiLSTM, and ResNet1D as heterogeneous base models to capture local morphological features, local-temporal fused features, and deep residual features of ECG signals, respectively. Following the Stacking ensemble strategy, the outputs of base models are taken as meta-features and fed into a logistic regression meta-classifier to obtain the final classification decision. Meanwhile, Z-score normalization and class weight method are adopted for data preprocessing to alleviate amplitude differences and class imbalance. Experimental results on the PTB myocardial infarction dataset and MIT-BIH arrhythmia dataset show that the CBR-Stacking model achieves accuracies of 99.55% and 98.86% in binary classification and five-class classification tasks, respectively, outperforming single base models and traditional machine learning methods in all evaluation metrics. Moreover, the time consumption of the model ensemble stage accounts for less than 0.1%, ensuring stable running efficiency. The proposed model effectively integrates the complementary advantages of multi-architecture deep learning models, improving the accuracy, robustness, and generalization ability of ECG classification, and provides an efficient and reliable solution for intelligent auxiliary diagnosis of electrocardiograms.

Keywords: ECG classification; Stacking; CNN; LSTM.

1. Introduction

Cardiovascular disease is one of the leading causes of death worldwide and has become a major public health concern threatening human health[1]. According to 2022 statistics from the World Health Organization, approximately 19.8 million people died from cardiovascular diseases globally, accounting for 32% of all deaths, of which 85% were caused by heart disease and stroke[2]. Most cardiovascular diseases can be prevented at an early stage through lifestyle modifications and risk factor management; therefore, non-invasive and efficient early detection technologies are crucial to reducing the incidence and mortality of such diseases.

Electrocardiography (ECG) records the electrical activity of the heart to reflect myocardial contraction and relaxation, serving as a core non-invasive tool for the early screening and clinical diagnosis of cardiovascular diseases such as arrhythmia and myocardial infarction[3]. Morphological and temporal features contained in ECG signals, including the QRS complex, P wave, and T wave, provide important evidence for disease diagnosis. However, traditional ECG diagnosis relies heavily on the professional experience of cardiologists, which is not only time-consuming and labor-intensive but also susceptible to subjective judgment, professional competence, and workload. In the context of medical informatization, massive ECG data have emerged, and the efficiency and accuracy of manual diagnosis can hardly meet the clinical demands of large-scale screening and rapid diagnosis. There is an urgent need to develop intelligent automatic ECG classification models.

In recent years, deep learning has been widely applied in ECG classification due to its end-to-end feature extraction capability. Convolutional Neural Networks (CNN) can effectively capture local morphological features of ECG signals[4]; Recurrent Neural Networks (RNN), especially Long Short-Term Memory (LSTM) networks, can model temporal dependencies between heartbeats [5]; attention mechanisms can strengthen the model's attention to diagnostically critical regions[6]; and Residual Networks (ResNet) alleviate the gradient degradation problem of deep networks[7]. Nevertheless, existing studies still suffer from obvious limitations: single models can hardly simultaneously extract both morphological and temporal features of ECG signals, leading to insufficient representation of complex arrhythmia patterns[8]. Moreover, medical ECG data generally suffer from class imbalance, and most models exhibit limited robustness and balanced classification performance under such conditions[9]. In addition, evaluation settings vary across different architectures[10], and the fusion of their complementary advantages has not been systematically investigated.

Ensemble learning improves overall prediction performance by integrating outputs from multiple base models, effectively compensating for the deficiencies of single models. As a hierarchical ensemble framework, Stacking utilizes two levels of learning—base model layer and meta-learner layer—to fully exploit the complementary strengths of different models. Accordingly, this paper proposes the CBR-Stacking deep ensemble learning algorithm, which integrates the feature extraction capabilities of three heterogeneous deep learning models: 1D-CNN, CNN-BiLSTM, and ResNet1D. Data preprocessing strategies are adopted to reduce amplitude discrepancies and class imbalance. Finally, experiments on public ECG datasets verify the effectiveness and efficiency of the proposed model, providing technical support for intelligent auxiliary ECG diagnosis.

2. Related Technologies

2.1. Stacking Ensemble Learning

Stacking ensemble learning is a hierarchical model fusion framework whose structure is shown in Fig.1. It efficiently integrates the prediction results of multiple base models through a two-layer learner structure, so as to improve the overall prediction performance and generalization ability[11]. Its core process consists of two stages.

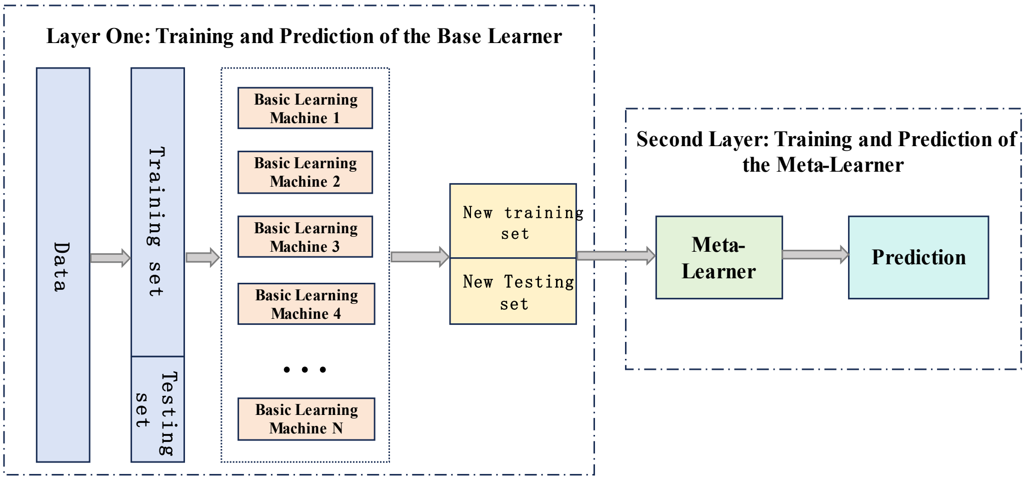

Fig.1 Stacking ensemble learning framework

The first layer is the base learner layer, which is composed of multiple base learners. Each base learner is trained and predicted on the original training set, and the output results are concatenated into a new feature set (i.e., "meta-features"), forming a new training set and a new prediction set for the second-layer learning.

The second layer is the meta-learner layer, which takes the meta-features output by the base learners as input, trains a high-level classifier to learn how to optimally fuse the prediction information of each base model, and finally outputs the global prediction result.

The selection of base learners emphasizes diversity and complementarity (such as classifiers of different paradigms) to fully capture the complex patterns in the data. The meta-learner can flexibly adopt linear models, tree models or other classifiers, which is responsible for the final decision fusion.

2.2. Convolutional Neural Network

Convolutional Neural Network (CNN)[12] is a type of feedforward neural network designed specifically for processing grid-structured data. Its core lies in effectively extracting local features from input data through the mechanisms of local connection and weight sharing. The basic structure of CNN is shown in Fig.2. For one-dimensional temporal data, one-dimensional Convolutional Neural Network (1D-CNN) extracts features along the time dimension using a sliding convolution kernel, which can effectively capture the local structure and morphological characteristics of the data.

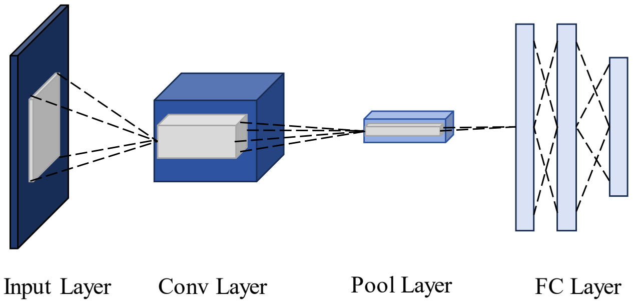

Fig.2 Structure of CNN

(1) Convolutional Layer: One-dimensional convolution can be formally defined as the sliding dot product operation between the input sequence and the convolution kernel. Let the input sequence be $x = [x_1, x_2, \dots, x_L] \in \mathbb{R}^L$ and the convolution kernel be $w = [w_1, w_2, \dots, w_K] \in \mathbb{R}^K$ where K is the kernel size, usually satisfying $K \ll L$. Then the feature value h_i output at the i-th position by the convolution operation can be expressed as:

$$h_i = f\left(\sum_{j=1}^{K} w_j \cdot x_{i+j-1} + b\right) \quad (1)$$

where b is the bias term, and $f(\cdot)$ is the nonlinear activation function, typically the Rectified Linear Unit (ReLU):

$$\mathrm{f(z)} = \mathrm{ReLU(z)} = \max(0, \mathrm{z}) \quad (2)$$

(2) Pooling Layer: A pooling layer is usually employed after the convolutional layer for feature dimensionality reduction, which retains key information while reducing the number of parameters and computational cost. Max Pooling outputs the maximum value within a local window, formally defined as:

$$p_i = \underset{j=1}{\overset{P}{max}} \left(h_{i+j-1}\right) \quad (3)$$

where P is the size of the pooling window.

(3) Fully Connected Layer and Classification: After multi-layer convolutional and pooling feature extraction, the network flattens the feature maps into a one-dimensional feature vector, which is fed into the fully connected layer for feature fusion, and classification is realized through the output layer.

For binary classification tasks, the Sigmoid activation function is used:

$$\hat{y} = \sigma\left(W_f \cdot h_g + b_f\right) = \frac{1}{1+e^{-\left(W_f \cdot h_g + b_f\right)}} \quad (4)$$

For multi-class classification tasks, the Softmax activation function is used to output the class probability distribution:

$$\widehat{y_k} = \frac{exp\left(W_{f,k} \cdot h_g + b_{f,k}\right)}{\sum_{j=1}^{C} exp\left(W_{f,j} \cdot h_g + b_{f,j}\right)} \quad (5)$$

where h_g is the global feature vector, W_f and b_f are the weight and bias of the fully connected layer, and C is the total number of classes.

2.3. Bidirectional Long Short-Term Memory Network

Long Short-Term Memory (LSTM) is an improved recurrent neural network, which effectively solves the problems of vanishing gradient and exploding gradient in traditional RNN when processing long sequences through gating mechanisms[13]. Bidirectional LSTM (BiLSTM) further utilizes both forward and backward information of the sequence to improve the ability to model temporal dependencies[14]. The basic structure of BiLSTM is shown in Fig.3.

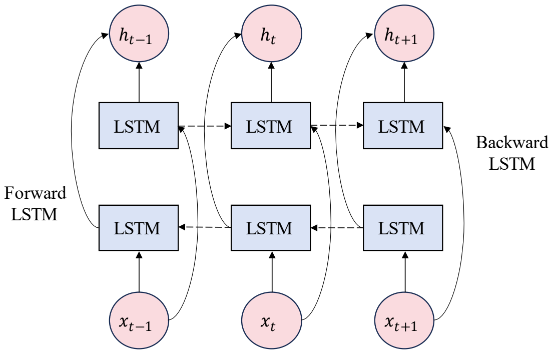

Fig.3 Structure of BiLSTM

LSTM controls information flow through forget gate, input gate, and output gate, and uses cell state to preserve long-term information. Given the current input x_t and the previous hidden state h_{t-1}, the update process is as follows:

Forget gate:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (6)$$

Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (7)$$

Candidate cell state:

$$\tilde{c_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (8)$$

Cell state update:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \quad (9)$$

Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (10)$$

Hidden state output:

$$h_t = o_t \odot tanh(c_t) \quad (11)$$

where $\sigma(\cdot)$ is the Sigmoid function, $tanh(\cdot)$ is the hyperbolic tangent function, ʘ denotes element-wise multiplication, W_f,W_i,W_c,W_o are the weight matrices of the corresponding gates, and b_f,b_i,b_c,b_o are the corresponding bias terms.

BiLSTM employs both forward LSTM and backward LSTM to capture complete contextual information. The final output is obtained by concatenating the forward and backward hidden states:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \quad (12)$$

or by element-wise summation:

$$h_t = \overrightarrow{h_t} + \overleftarrow{h_t} \quad (13)$$

where $\oplus$ denotes vector concatenation, $\overrightarrow{h_t}$ is the hidden state of the forward LSTM at time t, and $\overleftarrow{h_t}$ is the hidden state of the backward LSTM at time t.

2.4. Residual Network

Residual Network (ResNet) solves the problem of gradient degradation in training deep neural networks by introducing skip connections, enabling deeper networks to effectively improve model performance[15]. In traditional deep networks, gradients tend to vanish during backpropagation, while residual connections provide a direct propagation path for gradients, making the network easier to optimize.

(1) Residual Block: The basic unit of ResNet is the residual block. The structure of the residual block is shown in Fig.4. Given an input x, the output y is obtained by adding the input directly to the residual mapping F:

$$y = F(x, \{W_i\}) + x \quad (14)$$

When the dimensions of the input and output do not match, a linear projection W_s is used to align the dimensions, and the output becomes:

$$y = F(x, \{W_i\}) + W_s \cdot x \quad (15)$$

For one-dimensional residual networks (ResNet1D), the residual mapping $F(x)$ usually consists of two one-dimensional convolution layers, batch normalization (BN), and ReLU activation function, formally expressed as:

$$F(x) = ReLU\left(BN\left(W_2 * ReLU\left(BN(W_1 * x + b_1)\right) + b_2\right)\right) \quad (16)$$

where $*$ denotes one-dimensional convolution, W_1 and W_2 are convolution weights, b_1 and b_2 are the corresponding bias terms, $BN(\cdot)$ is the batch normalization operation, and $ReLU(\cdot)$ is the rectified linear unit activation function.

(2) Gradient Propagation Property: The key advantage of residual connections lies in efficient gradient propagation. Let the loss function be L, the gradient with respect to the input x can be derived as:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \left(1 + \frac{\partial F}{\partial x}\right) \quad (17)$$

where the constant term 1 ensures that the gradient can propagate directly from the output layer to the input layer, significantly alleviating the vanishing gradient problem in training deep networks and making deep networks easier to optimize.

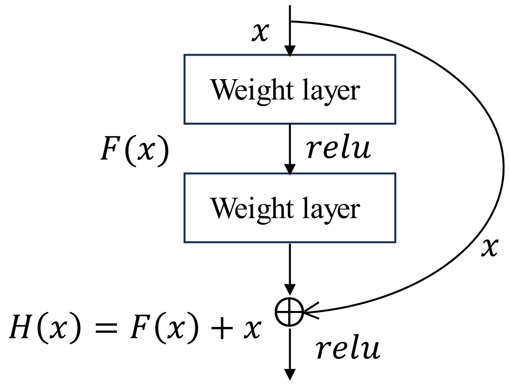

Fig.4 Structure of ResNet residual block

2.5. Attention Mechanism

The core of the attention mechanism is to assign higher weights to key information by quantifying the importance differences of input features, so as to realize adaptive weighted fusion of features. Compared with traditional feature fusion methods, this mechanism can effectively focus on task-related core information and suppress irrelevant noise interference. Its core execution process can be divided into three stages: score calculation, weight normalization, and weighted fusion, and its basic structure is shown in Fig.5.

The attention mechanism relies on three key components: Query, Key, and Value. Its basic principle is described as follows:

(1) Attention Score Calculation

The attention score is used to quantify the correlation between the Query vector and each Key vector, reflecting the contribution of corresponding information to the target task. Let the query vector be Q and the set of key vectors be $K = [K_1, K_2, \dots K_n]$, where n is the number of input items. The attention score s_i of the i-th key vector with respect to the query vector is calculated by the scoring function $f(Q, K_i)$:

$$s_i = f(Q, K_i) \tag{18}$$

Common scoring functions include dot-product, scaled dot-product, and additive models, which can be flexibly selected according to feature dimensions and task complexity.

(2) Attention Weight Normalization

To convert attention scores into a usable probability distribution, all scores are normalized by the Softmax function to ensure the sum of weights is 1. For the obtained attention score s_i, the normalized weight of the i-th item is computed as:

$$\alpha_i = Softmax(s_i) = \frac{exp(s_i)}{\sum_{j=1}^{n} exp(s_j)} \tag{19}$$

(3) Weighted Fusion to Generate Attention Value

The normalized attention weights are multiplied element-wise with the corresponding Value vectors, and all weighted results are summed to obtain the final attention value that integrates key information. Let the set of value vectors be $V = [V_1, V_2, \dots V_n]$. The attention value C is given by:

$$C = \sum_{i=1}^{n} \alpha_i \cdot V_i \tag{20}$$

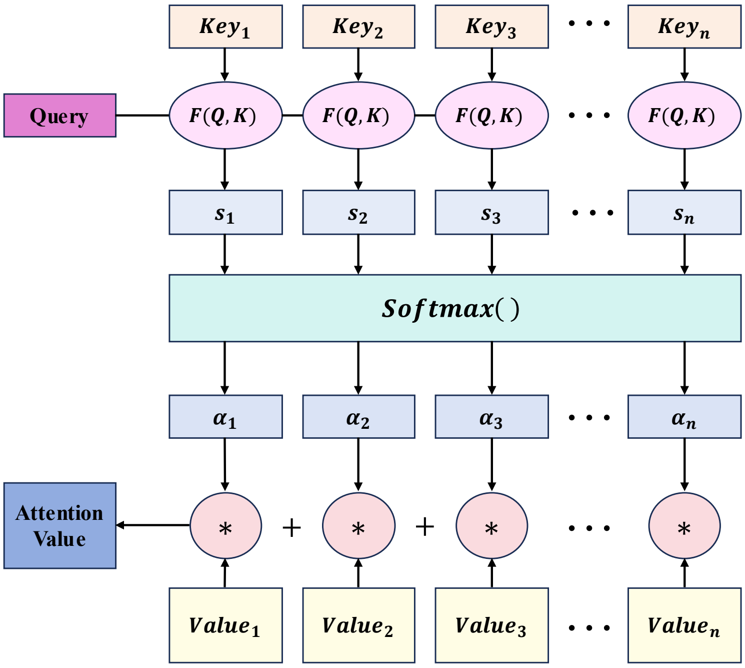

Fig.5 Structure of the attention mechanism

3. Construction of the CBR-Stacking Model

This paper proposes the CBR-Stacking ensemble framework, which aims to integrate the advantages of different deep learning architectures to improve the performance of ECG signal classification. Considering the morphological and temporal characteristics of electrocardiogram (ECG), three complementary basic models are first constructed. The CNN model efficiently extracts local morphological features through cascaded convolutional blocks, focusing on short-range structural patterns of heartbeats. The CNN-BiLSTM model introduces a bidirectional long short-term memory network based on convolutional features to realize bidirectional temporal modeling and capture long-range dependencies and contextual information among heartbeat sequences. The ResNet-1D model adopts a one-dimensional residual connection structure to support stable training of deeper networks, alleviate the vanishing gradient problem, and achieve hierarchical representation of complex ECG patterns. On this basis, the Stacking ensemble strategy is introduced, in which the outputs of the three basic models are used as meta-features and fed into a meta-learner to complete the final classification decision. This fully integrates the complementary advantages of each architecture and enhances the robustness and generalization ability of the model under imbalanced data. Overall, the framework provides a systematic implementation path for exploring the synergistic effect of multi-branch feature extraction and model fusion in ECG signal classification.

3.1. 1D-CNN Model

The CNN model constructed in this paper is a one-dimensional convolutional neural network for ECG signal classification tasks, adapted to two types of ECG datasets: PTB (binary classification) and MIT-BIH (5-class classification). It is mainly used to extract local morphological features from 187-dimensional ECG time-series signals. The 1D-CNN model diagram is shown in Fig.6.

The network consists of eight convolutional layers forming four feature extraction modules.

The first module contains two convolutional layers with 16 kernels of size 5×1, which directly act on the original ECG signal with dimension (187,1) to extract primary local features. After 2×1 max pooling, Dropout with a rate of 0.1 is applied to suppress overfitting;

The second module includes two convolutional layers with 32 kernels of size 3×1 to encode mid-level local features, followed by repeated pooling and Dropout (0.1);

The third module uses two convolutional layers with 32 kernels of size 3×1 to further deepen feature extraction. After pooling and Dropout (0.1), it focuses on key waveform patterns;

The fourth module contains two convolutional layers with 256 kernels of size 3×1 to complete deep feature encoding. Global max pooling is used to compress variable-length temporal features into fixed-dimensional vectors, and Dropout with a rate of 0.2 is adopted for feature selection.

After the feature extraction stage, two fully connected layers with 64 neurons perform fusion and mapping of global features. The output layer adapts different activation functions and loss functions according to the classification task of the dataset. For the PTB binary classification task: the output layer uses the Sigmoid activation function to calculate the probability of a single category, and the loss function adopts binary cross-entropy. For the MIT-BIH 5-class classification task: the output layer uses the Softmax activation function to calculate the multi-class probability distribution, and the loss function adopts sparse categorical cross-entropy.

The model is trained with the Adam optimizer (initial learning rate 0.001), integrated with three callback strategies: ModelCheckpoint (saving the optimal weight of the validation set accuracy), EarlyStopping (terminating training if the validation set accuracy does not improve for 5 epochs), and ReduceLROnPlateau (reducing the learning rate if the validation set accuracy does not improve for 3 epochs). The maximum number of training iterations is 100. This architecture captures local morphological features such as QRS complexes, P waves, and T waves in ECG signals at multiple scales, providing basic spatial feature representation capabilities for the CBR-Stacking ensemble framework.

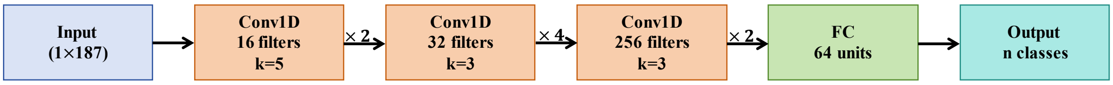

Fig.6 1D-CNN model

3.2. CNN-BiLSTM Model

The CNN-BiLSTM hybrid architecture constructed in this paper realizes the joint modeling of local and temporal features of ECG signals through the combined strategy of "CNN extracting local morphological features + BiLSTM capturing bidirectional temporal dependencies". The CNN-BiLSTM model diagram is shown in Fig.7.

The core structure and training characteristics of the model are as follows:

CNN module: composed of 3 groups of convolution-batch normalization-pooling units (16 5×1 convolution kernels + 2 groups of 32 3×1 convolution kernels), Dropout (0.1) is introduced in each layer to output low-dimensional temporal feature sequences;

BiLSTM module: adopts a bidirectional LSTM layer with 64 hidden units to perform bidirectional temporal encoding on the output sequence of the CNN. After outputting fixed-dimensional feature vectors, Dropout (0.2) regularization is applied;

Classification layer: connected to a fully connected layer with 64 neurons (including batch normalization + Dropout 0.2). For the PTB binary classification task, Sigmoid activation and binary cross-entropy loss are used; for the MIT-BIH 5-class classification task, Softmax activation and sparse categorical cross-entropy loss are used. Class weights are introduced in both tasks to alleviate data imbalance.

The Adam optimizer (initial learning rate 0.001) is uniformly used, integrated with three callback strategies: ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau (maximum number of iterations is 100).

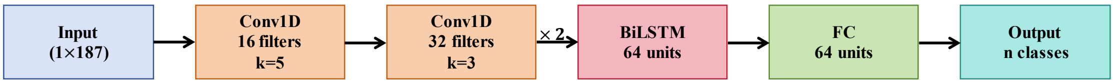

Fig.7 CNN-BiLSTM model

3.3. ResNet1D Model

The ResNet1D model constructed in this paper is based on a one-dimensional residual network architecture, which alleviates the vanishing gradient problem of deep networks through residual connections and realizes deep feature learning of ECG signals. The ResNet1D model diagram is shown in Fig.8.

The core structure and training characteristics of the model are as follows:

Residual block module: the custom residual block is composed of 2 layers of convolution-batch normalization units (default 3×1 convolution kernels). The shortcut path completes dimension alignment through 1×1 convolution kernels, and finally fuses the features of the main path and the shortcut path through the Add layer to realize the retention and enhancement of shallow and deep features;

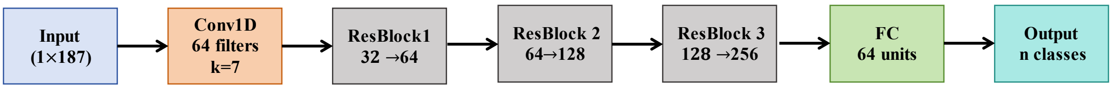

Fig.8 ResNet1D model

Backbone network: the initial convolutional layer (16 5×1 convolution kernels + batch normalization + 2×1 pooling) extracts primary features, followed by 3 stacked residual blocks (output channels 32, 64, 128). Each residual block is followed by 2×1 max pooling to gradually improve the feature dimension;

Classification layer: after being compressed into fixed-dimensional vectors by global max pooling, it is connected to a fully connected layer with 64 neurons (including Dropout 0.2). For the PTB binary classification task, Sigmoid activation and binary cross-entropy loss are used; for the MIT-BIH 5-class classification task, Softmax activation and sparse categorical cross-entropy loss are used. Class weights are introduced in both tasks to alleviate data imbalance.

The model training adopts the Adam optimizer (initial learning rate 0.001), and integrates the same three callback strategies as the 1D-CNN and CNN-BiLSTM models (ModelCheckpoint, EarlyStopping, ReduceLROnPlateau), with a maximum of 100 training epochs. The residual connection structure enables the network to stably extract deep-level ECG features, avoiding the degradation of model performance caused by the increase of network depth, and providing deep feature support for the CBR-Stacking ensemble framework.

3.4. CBR-Stacking Ensemble Strategy

The CBR-Stacking ensemble framework adopts a two-layer hierarchical structure, which is divided into a base model layer (CBR layer) and a meta-learner layer, and the specific implementation process is consistent with the flowchart shown in Fig.9.

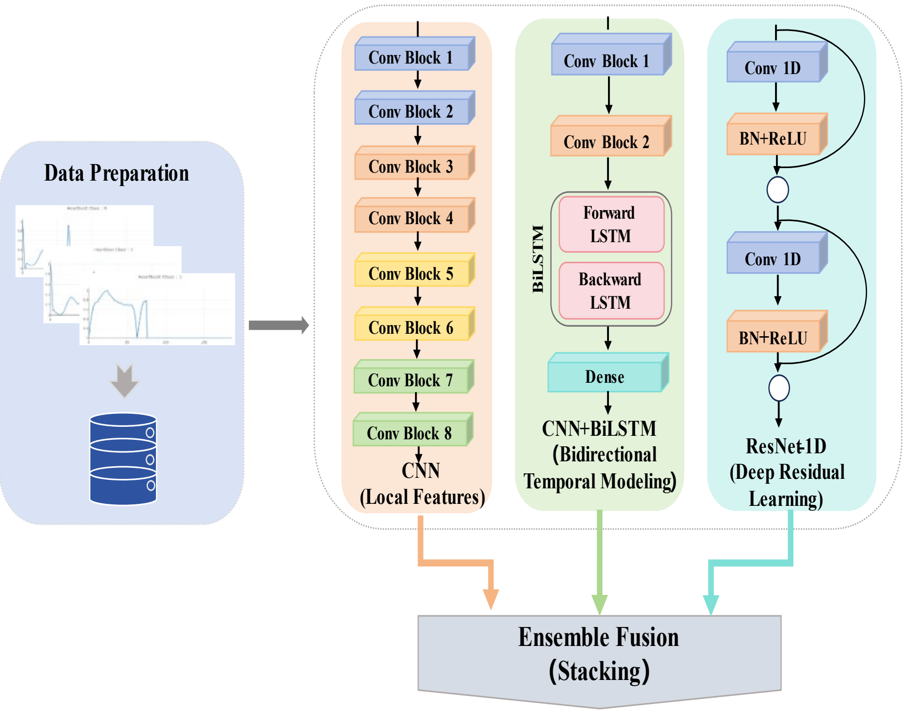

Fig.9 CBR-Stacking Model

3.4.1. Base Model Layer (CBR Layer)

The base model layer is composed of the three heterogeneous deep learning models constructed in Sections 3.1, 3.2, and 3.3 (1D-CNN, CNN-BiLSTM, ResNet1D), which are responsible for feature extraction and preliminary classification of ECG signals. The core design idea is to rely on the complementary advantages of different models to comprehensively capture the multi-dimensional features of ECG signals: 1D-CNN focuses on local morphological features, CNN-BiLSTM focuses on local-temporal fused features, and ResNet1D focuses on deep residual features.

In the training stage of the base model layer, the three models are trained independently on the preprocessed training set respectively, and the callback strategies and training parameters (optimizer, learning rate, number of epochs, etc.) are kept consistent to ensure the fairness of model training and the comparability of output results. After the training of each base model is completed, the output results (preliminary classification probability or category prediction) of the model on the training set and the test set are extracted respectively, and these output results are spliced in sequence to form a new meta-feature set. The meta-feature set is used as the input data of the meta-learner layer to provide a comprehensive feature basis for the final classification decision.

3.4.2. Meta-learner Layer

The meta-learner layer is the core of the Stacking ensemble strategy, which is responsible for learning the optimal fusion method of the output results of the base model layer and completing the final classification decision. Considering the characteristics of the meta-feature set (low dimension, clear interpretability) and the requirements of classification accuracy and training efficiency, this paper selects the logistic regression model as the meta-learner.

The logistic regression meta-learner takes the meta-feature set output by the base model layer as input, and trains on the meta-feature set corresponding to the training set to learn the weight distribution of each base models output. In the prediction stage, the meta-learner inputs the meta-feature set of the test set, integrates the preliminary classification information of the three base models according to the learned weight distribution, and outputs the final classification result.

Compared with other meta-learners (such as decision trees, random forests), the logistic regression model has the advantages of simple structure, fast training speed, and strong interpretability, which can effectively avoid overfitting caused by complex meta-learners, and ensure the efficiency and stability of the ensemble framework while improving the classification accuracy.

4. Experimental Results and Analysis

4.1. Data Sources

In this paper, two public ECG datasets, PTB and MIT-BIH, are adopted for experiments, both of which are standard clinical ECG recordings. The detailed descriptions of the datasets are shown in Table 1.

(1) PTB Myocardial Infarction Dataset. The PTB dataset consists of ECG recordings from 290 subjects, including 148 patients diagnosed with myocardial infarction, 52 healthy controls, and the remaining subjects diagnosed with 7 other diseases. In this study, only data from the myocardial

infarction group and the healthy control group are used for experiments.

(2) MIT-BIH Arrhythmia Dataset. This dataset is composed of ECG recordings from 47 subjects, and each heartbeat is annotated by at least two cardiologists. According to the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard, the heartbeats in the dataset are divided into five distinct classes.

Table 1. Dataset Description

Dataset	Sample Size	Sampling Rate	Feature Dimension	Label
PTB	14552	125 Hz	187	0 = Normal
				1 = Abnormal
MIT-BIH	109446	125 HZ	187	0 = N (Normal beat)
				1 = S (Supraventricular ectopic beat)
				2 = V (Ventricular ectopic beat)
				3 = F (Fusion beat)
				4 = Q (Unknown beat)

4.2. Data Preprocessing Strategy

To solve the problems of amplitude difference between ECG signals and class imbalance in the dataset, which affect the training effect and classification performance of the model, this paper designs a two-step data preprocessing strategy to optimize the input data of the CBR-Stacking model, ensuring the stability and reliability of model training.

4.2.1. Z-score Normalization

Due to differences in recording equipment, recording environment, and individual physical conditions of patients, there are obvious amplitude differences between different ECG signals, which will interfere with the feature extraction effect of the model. Therefore, this paper adopts Z-score normalization to standardize the ECG signal data, converting the signal amplitude to a unified standard range.

The Z-score normalization process is to calculate the mean value and standard deviation of the ECG signal sequence, and then normalize each data point in the sequence to eliminate the influence of amplitude difference. After normalization, the ECG signal follows a standard normal distribution, which is conducive to the rapid convergence of the model during training and improves the consistency of feature extraction.

4.2.2. Class Weight Method

The PTB myocardial infarction dataset and MIT-BIH arrhythmia dataset both have obvious class imbalance problems: the number of normal ECG samples is much larger than that of abnormal samples, which will lead to the model being biased towards the majority class during training, resulting in low classification accuracy of the minority class (abnormal ECG).

To alleviate this problem, this paper introduces the class weight method in the model training process. According to the sample size of each category in the dataset, the weight of each category is calculated adaptively: the weight of the minority class is set higher, and the weight of the majority class is set lower. When calculating the loss function, the model will assign higher weights to the misclassified samples of the minority class, thereby improving the attention of the model to the minority class, balancing the training process of the model, and enhancing the classification robustness of the model under imbalanced data.

4.3. Analysis of Prediction Results for Base Models and CBR-Stacking

To verify the effectiveness of the proposed CBR-Stacking ensemble model, comparative experiments were conducted on the PTB and MIT-BIH datasets, respectively, using three base models (1D-CNN, CNN-BiLSTM, ResNet1D) and the CBR-Stacking model. Five evaluation metrics were adopted: Accuracy (ACC), Precision (PRE), Recall (REC), F1-score, and AUC. The results are presented in Table 2 and Table 3.

The experimental results show that the CBR-Stacking model outperforms all single base models in all evaluation metrics on both datasets. This indicates that integrating the feature and decision-making advantages of different base models can effectively improve the accuracy and robustness of ECG signal classification, demonstrating stronger reliability and generalization ability in both binary and multi-classification tasks.

Table 2. Prediction Results of Base Models and CBR-Stacking on the PTB Dataset

Model	ACC	PRE	REC	F1	AUC
CNN	0.9935	0.9967	0.9943	0.9955	0.9971
CNN-BiLSTM	0.9863	0.9886	0.9924	0.9905	0.9970
ResNet1D	0.9835	0.9904	0.9867	0.9886	0.9962
CBR-Stacking	**0.9955**	**0.9967**	**0.9971**	**0.9969**	**0.9987**

Table 3. Prediction Results of Base Models and CBR-Stacking on the MIT-BIH Dataset

Model	ACC	PRE	REC	F1	AUC
CNN	0.9846	0.9420	0.8839	0.9841	0.9955
CNN-BiLSTM	0.9782	0.8626	0.9437	0.9632	0.9956
ResNet1D	0.9762	0.8974	0.9486	0.9692	0.9956
CBR-Stacking	**0.9886**	**0.9739**	**0.9421**	**0.9822**	**0.9972**

Fig.10 and 11 show the confusion matrices obtained by the CBR-Stacking model on the PTB and MIT-BIH datasets, respectively. The confusion matrices clearly demonstrate that the proposed ensemble model achieves excellent recognition performance for both majority and minority classes, with stable and reliable classification results, further validating its effectiveness in practical ECG classification tasks.

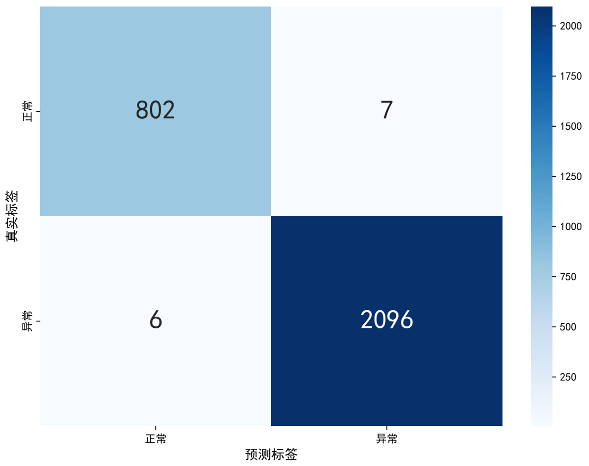

Fig.10 Confusion Matrix on the PTB Dataset

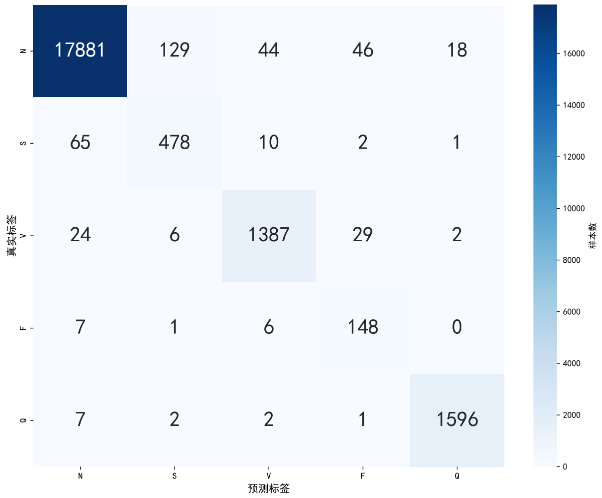

Fig.11 Confusion Matrix on the MIT-BIH Dataset

4.4. Comparative Experiments

To further validate the effectiveness of the proposed CBR-Stacking model, comparative experiments were conducted on both the PTB and MIT-BIH datasets, comparing the model with various traditional machine learning methods and recent related research approaches. The evaluation metrics used were ACC, PRE, REC, and F1. The results are shown in Table 4 and Table 5.

Table 4. Prediction Results of Various Models and CBR-Stacking on the PTB Dataset

Methods	ACC (%)	PRE (%)	REC (%)	F1 (%)
Gaussian Naive Bayes	61.1	65.0	68.0	60.0
LR	82.2	79.0	75.0	76.0
Decision trees	92.1	90.0	90.0	90.5
SVM	94.48	92.0	95.0	93.0
KNN	92.2	90.0	91.0	90.5
RF	96.9	97.0	95.5	96.0
Sharma et al.[9]	95.63	99.03	92.73	95.77
Kumar et al. [10]	98.66	98.92	93.88	96.34
CBR-Stacking	**99.55**	**99.67**	**99.71**	**99.69**

Table 5. Prediction Results of Various Models and CBR-Stacking on the MIT-BIH Dataset

Methods	ACC (%)	PRE (%)	REC (%)	F1 (%)
Gaussian Naive Bayes	67.0	67.4	67.0	66.8
LR	67.42	67.4	68.0	67.4
Decision trees	94.4	94.4	94.2	94.4
SVM	96.18	96.4	96.2	96.0
KNN	97.22	97.4	97.2	97.2
RF	98.24	98.4	98.2	98.2
Sharma et al.[9]	95.63	99.03	92.73	95.77
Kumar et al. [10]	98.66	98.92	93.88	96.34
CBR-Stacking	**98.86**	**97.39**	**94.21**	**98.22**

As shown in Table 4, on the PTB dataset, the proposed method outperforms traditional machine learning algorithms and the compared literature methods in all evaluation metrics. On the MIT-BIH dataset (Table 5), the CBR-Stacking model also achieves superior classification performance. The experimental results demonstrate that the proposed ensemble learning framework can effectively improve ECG signal classification performance, with higher reliability and generalization ability.

4.5. Computational Efficiency of CBR-Stacking

To verify the execution efficiency of the CBR-Stacking model, this study tested and recorded the models runni ng time on the PTB and MIT-BIH datasets under a hardware environment with an Intel Core i7-12700H processor and 32GB of RAM, and a software environment with Python 3.9. The running time is divided into two parts: the base model training and prediction phase, and the ensemble phase. The specific results are shown in Table 6.

Table 6. Training Time of CBR-Stacking on Different Datasets

Dataset	Model	Training Time (s)	Ensemble Time (s)	Total Time (s)
PTB	CNN	224.64	1.34	2506.38
	CNN-BiLSTM	1733.83		
	ResNet1D	546.57		
MIT-BIH	CNN	1273.99	1.70	3360.52
	CNN-BiLSTM	764.92		
	ResNet1D	1319.91		

The experimental results show that the total running time of the CBR-Stacking model on the PTB dataset is 2506.38 s, of which the base model training and prediction phase takes 2505.04 s, and the ensemble phase takes only 1.34 s. On the MIT-BIH dataset, the total running time is 3360.52 s, with the base model training and prediction phase taking 3358.82 s and the ensemble phase taking 1.70 s. It can be seen that the overall time consumption of the model mainly comes from the training process of each base model, with the ensemble phase accounting for less than 0.1% of the total time. The model also exhibits stable running efficiency on both datasets.

These results provide a reliable time dimension reference for the practical clinical application of the CBR-Stacking model.

5. Conclusion

This chapter proposes the CBR-Stacking deep ensemble learning algorithm for ECG classification. The algorithm consists of three modules: data preprocessing, multi-branch feature extraction and modeling, and Stacking ensemble fusion. By constructing three heterogeneous base models (1D-CNN, CNN-BiLSTM, ResNet1D) to extract different dimensional features of ECG signals, and then using a logistic regression meta-classifier to fuse the outputs of each model for final classification. Using the PTB and MIT-BIH public ECG datasets as experimental subjects, multiple sets of comparative experiments were conducted after data preprocessing. The results show that the CBR-Stacking algorithm achieves excellent performance in ECG classification prediction, effectively alleviating the problem of data imbalance and improving the accuracy, robustness, and generalization ability of ECG signal classification. It provides an efficient and reliable solution for intelligent ECG classification.

References

[1] National Center for Cardiovascular Diseases, Writing Group of Chinese Cardiovascular Health and Diseases Report, Hu, S. H. Summary of Chinese Cardiovascular Health and Diseases Report 2023[J]. Chinese Circulation Journal, 2024, 39(7): 625-660.

[2] World Health Organization, Cardiovascular diseases (CVDs). [Online]. Available: https://www.who.int/news-room/factsheets/detail/cardiovascular-diseases-(cvds)

[3] Tran T L, Thai B T, Tran V K, et al. Real-time Assessment of ECG Classification based on Time-series Data and Other Types of Features[J]. Journal of Engineering & Technological Sciences, 2025, 57(4).

[4] Jasvitha B D, Kanagaraj K, Murali K, et al. 1D CNN Framework on ECG Signals[J]. 2024 3rd International Conference for Innovation in Technology (INOCON), 2024:1-6.

[5] Jyotishi, Debasish, and Samarendra Dandapat. An LSTM-based model for person identification using ECG signal. IEEE Sensors Letters 4.8 (2020): 1-4.

[6] Wang, J. R., Yu, X. X., Xin, B. B., Gao, R. F., Li, G. Research on ECG Classification Algorithm Based on Improved U-Net Network[J]. Journal of Chongqing University of Technology (Natural Science), 2024, 38(1): 142-149.

[7] Wang, G. J., Wu, T., Wang, L., et al. Research on Electrocardiogram Diagnosis Based on Machine Learning[J]. Journal of Practical Electrocardiology, 2020, 29(4): 262-268+2977.

[8] Rai H M, Chatterjee K, Dashkevych S. The prediction of cardiac abnormality and enhancement in minority class accuracy from imbalanced ECG signals using modified deep neural network models[J]. Computers in Biology and Medicine, 2022, 150: 106142.

[9] Sharma P, Dinkar S K, Gupta D. A novel hybrid deep learning method with cuckoo search algorithm for classification of arrhythmia disease using ECG signals[J]. Neural Computing and Applications, 2021, 33: 13123-13143.

[10] Kumar S, Mallik A, Kumar A, et al. Fuzz-ClustNet: Coupled fuzzy clustering and deep neural networks for Arrhythmia detection from ECG signals[J]. Computers in Biology and Medicine, 2023, 153: 106511.

[11] Liu, H. D., Sun, X. H., Wang, B., et al. Research on Poverty Return Risk Prediction of Poverty-Alleviated Population Based on Stacking Ensemble Learning[J]. Journal of Hebei Agricultural University, 2024, 47(6): 75-82.

[12] Ige A O, Sibiya M. State-of-the-art in 1D Convolutional Neural Networks: A survey[J]. IEEE Access, 2024.

[13] S.-N. Tang, Y.-H. Chen, Y.-W. Chang, et al. Hybrid CNN-LSTM Network for ECG Classification and Its Software-Hardware Co-Design Approach. 2023 20th International SoC Design Conference (ISOCC), 2023, pp. 173-174.

[14] Basora M G, Mulayim M O. An Attention-Augmented VAE-BiLSTM Framework for Anomaly Detection in 12-Lead ECG Signals[J]. 2025.

[15] N. ZALIGA, S. ELOUAHAM and A. DLIOU. 1 Dimensional Residual Network for Ecg Signal Denoising. 2025 International Conference on Circuit, Systems and Communication (ICCSC). 2025, pp. 1-5.