Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6648432, 17 pages
https://doi.org/10.1155/2021/6648432

Research Article

Real-Time Arrhythmia Classification Algorithm Using Time-Domain ECG Feature Based on FFNN and CNN

Jing Cai,[1] **Ge Zhou**,[1] **Mengkun Dong**,[1] **Xinlei Hu**,[1] **Guangda Liu**,[1] **and Weiguang Ni**[1,2]

[1]*College of Instrumentation and Electrical Engineering, Jilin University, Changchun 130026, China*
[2]*Physical Education College, Jilin University, Changchun 130012, China*

Correspondence should be addressed to Guangda Liu; gdliu@jlu.edu.cn

Received 27 December 2020; Revised 23 April 2021; Accepted 6 May 2021; Published 17 May 2021

Academic Editor: Zahid Mehmood

To solve the problem of real-time arrhythmia classification, this paper proposes a real-time arrhythmia classification algorithm using deep learning with low latency, high practicality, and high reliability, which can be easily applied to a real-time arrhythmia classification system. In the algorithm, a classifier detects the QRS complex position in real time for heartbeat segmentation. Then, the ECG_RRR feature is constructed according to the heartbeat segmentation result. Finally, another classifier classifies the arrhythmia in real time using the ECG_RRR feature. This article uses the MIT-BIH arrhythmia database and divides the 44 qualified records into two groups (DS1 and DS2) for training and evaluation, respectively. The result shows that the recall rate, precision rate, and overall accuracy of the algorithm's interpatient QRS complex position prediction are 98.0%, 99.5%, and 97.6%, respectively. The overall accuracy for 5-class and 13-class interpatient arrhythmia classification is 91.5% and 75.6%, respectively. Furthermore, the real-time arrhythmia classification algorithm proposed in this paper has the advantages of practicability and low latency. It is easy to deploy the algorithm since the input is the original ECG signal with no feature processing required. And, the latency of the arrhythmia classification is only the duration of one heartbeat cycle.

1. Introduction

According to the World Health Organization (WHO), 17.9 million people died of cardiovascular disease in 2019, of which 85% died of sudden heart disease and stroke [1]. Arrhythmia refers to a problem with the frequency or rhythm of the heartbeat, and severe arrhythmia may cause lethal heart disease [2]. Clinically, doctors usually diagnose it by analyzing the patient's electrocardiogram (ECG) with his/her relevant medical history and clinical manifestations [3]. However, abnormal ECG signals usually occur by chance, which cannot be obtained from a short-term ECG. It is time-consuming and labor-intensive and lacks objectivity to only rely on manual processing of a patient's long-term ECG records. Moreover, centralized analysis after recording ECG lacks real-time performance and cannot deal well with the sudden risk of patients.

With the development of computer science and technology, computer-aided diagnosis (CAD) for ECG analysis has helped solve the shortage of manual processing of ECG [4]. An increasing number of algorithms have been proposed for the automatic analysis of ECG signals for arrhythmia classification. Generally, an arrhythmia classification algorithm consists of four steps: preprocessing, heartbeat segmentation, feature extraction, and classification algorithm. Heartbeat segmentation has been studied for 30 years [5–8]. The classical heartbeat segmentation method uses an adaptive threshold method [5]. With the deepening of research, more and more new technologies have been applied to heartbeat segmentation algorithms, such as wavelet transform [9], genetic algorithm [10], and neural network [11]. The accuracy of heartbeat segmentation has a great impact on the final arrhythmia classification. However, many studies on arrhythmia classification algorithms directly use the heartbeat markers in the database, ignoring the influence of errors in heartbeat segmentation on the overall algorithm. Typical ECG features include single features

(such as heart rate variability, QRS width, PQ/PR interval, and amplitude of QRS) and time-domain features or corresponding frequency-domain features extracted directly from ECG signals. More popular arrhythmia classification algorithms include support vector machine (SVM) [12, 13], artificial neural network (ANN) [14, 15], linear discriminant (LD) [16], and logistic regression (LR) [17].

Deep learning is a branch of machine learning. It is a computational model with multiple processing layers to learn data representation with multiple levels of abstraction [18]. Deep learning optimizes the parameters of each layer by backpropagation and discovers complex structures in big datasets [18]. It has been proven to be useful for many disciplines, such as computer vision, speech recognition, natural language processing, and bioinformatics. Increasingly methods based on deep learning are used to study the classification of arrhythmia. Mathews et al. [19] proposed a deep learning-based ECG classifier using single-lead ECG and trained a deep learning model-based classifier to classify arrhythmias on the ECG signal with a 114 Hz sampling rate. Paweł and Acharya [20] used long-duration (10 s) ECG signal segments, strengthened the characteristic ECG signal features with spectral power density estimation, and introduced a novel three-layer deep genetic ensemble of classifiers. Shaker et al. [21] proposed a novel data augmentation technique using generative adversarial networks (GANs) to balance the dataset and effectively improve the performance of ECG classification over the same models trained on the original dataset. These studies have made good progress in the accuracy and the interpatient performance of the arrhythmia classification algorithm; however, they lack the real-time improvement of the algorithm. In practice, developing the most appropriate classifier that is capable of classifying arrhythmia in real time is also an issue in ECG arrhythmia classification [22].

Due to the specificity of individual ECGs, the main concern in practical application is the interpatient performance of the arrhythmia classification algorithm. There are two main ways to improve it. The first method is to use an expert to annotate arrhythmia on a portion of a specific patient's ECG and fine-tune the model with the annotation to improve the model performance for this patient [23–26]. Among them, Luo et al. [26] proposed a patient-specific arrhythmia classifier based on deep learning, in which a deterministic patient-specific heartbeat classifier is fine-tuned on heartbeat samples that include a small subset of individual samples (the overall accuracy increases from 89.3% to 97.5%). However, this method is feasible but not scalable because fine-tuning the model requires expert intervention. Another method is to train a general classifier with good interpatient performance through a reasonable selection of features, normalization, training datasets, and evaluation methods [14, 27–31]. The second method is used in this paper because it is cheaper and more practical.

The combination of edge computing [32] and wearable ECG acquisition technology for real-time arrhythmia monitoring cannot only help patients monitor their health and prevent sudden risks but also has the advantages of low latency, low power consumption, low bandwidth, and high privacy. An arrhythmia classification algorithm with low latency, high practicability, and reliability is the key to the monitoring system. Therefore, this paper proposes a real-time arrhythmia classification algorithm using deep learning. In the algorithm, a classifier based on the FFNN (Feedforward Neural Network) model first detects the QRS complex position for real-time heartbeat segmentation. Then, the time-domain morphological features of each heartbeat cycle are extracted according to the heartbeat segments, and another classifier based on the CNN (Convolutional Neural Network) model classifies arrhythmia in real time. The algorithm achieves a real-time performance of heartbeat cycle latency in arrhythmia classification and overcomes the shortcomings of traditional arrhythmia classification algorithms, which are unscalable and complex in feature processing and not suitable for edge computing. This paper provides a real-time, efficient, and reliable arrhythmia classification algorithm for edge computing-based ECG monitoring systems.

2. Materials and Methods

2.1. Data Description. The MIT-BIH arrhythmia database [33] is published by the Massachusetts Institute of Technology-Beth Israel Hospital. It contains 48 ECG records with a sampling rate of 360 Hz and a duration of 30 minutes. In this paper, only the ECG signals of the MLII lead are used for arrhythmia classification, which requires less hardware and computational cost, while giving a satisfactory overall accuracy [31, 34].

ANSI/AAMI [35] categorizes the 15 recommended classes of arrhythmia into five superclasses which are normal (N), supraventricular ectopic beat (SVEB), ventricular ectopic beat (VEB), fused beat (F), and unknown beats (Q). It is recommended to classify with only these superclasses. Since there are only 13 classes of arrhythmia in MIT-BIH, this paper designs the heartbeat classification model as a 13-class classification model, which can then get the 5-class classification results according to the class hierarchy. Some studies also classify arrhythmia into other five classes: normal (N), left bundle branch block (LBBB), right bundle branch block (RBBB), premature ventricular contraction (PVC), and atrial premature beat (APB) [36–38].

To better adapt to the actual environment and to improve the practicality of the algorithm, this paper divides the 44 records in the MIT-BIH database without pacemakers into two groups, according to the work of Chazal et al. [16]. The first group, named the DS1 group, consists of the records of 101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, and 230. The second group, named the DS2 group, consists of the records of 100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, and 234. The records of Groups DS1 and DS2 are basically from different patients (201 and 202 are from the same patient but in different groups). Figure 1 shows part of the waveform of each data record in the DS1 and DS2 groups.

Table 1 shows the hierarchy of superclasses and subclasses in arrhythmia classes. It also contains descriptions and quantity statistics of each class in Groups DS1 and DS2.

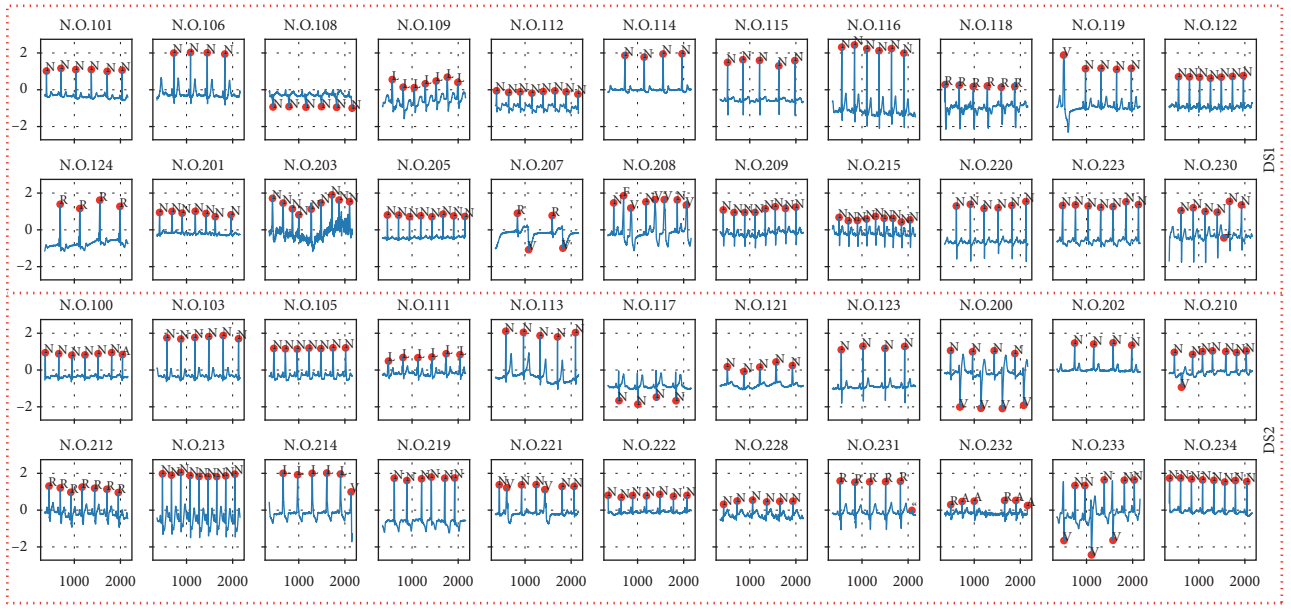

Figure 1: Part of the waveform of each data record in the DS1 and DS2 groups.

Table 1: Description and quantity statistics of each class in the MIT-BIH database.

Superclass labels and descriptions	All	ds0	ds1	Subclass labels and descriptions	All	ds0	ds1
(N) any heartbeat not categorized as SVEB, VEB, F, or Q	90125	45866	44259	(N) normal beat	74546	38102	36444
				(L) left bundle branch block beat	8075	3949	4126
				(R) right bundle branch block beat	7259	3783	3476
				(j) nodal (junctional) escape beat	229	16	213
				(e) atrial escape beat	16	16	0
(SVEB) supraventricular ectopic beat	2781	944	1837	(A) atrial premature beat	2546	810	1736
				(a) aberrated atrial premature beat	150	100	50
				(J) nodal (junctional) premature beat	83	32	51
				(S) supraventricular premature or ectopic beat	2	2	0
(VEB) ventricular ectopic beat	7009	3788	3221	(V) premature ventricular contraction	6903	3683	3220
				(E) ventricular escape beat	106	105	1
(F) fusion beat	803	415	388	(F) fusion of ventricular and normal beat	803	415	388
(Q) unknown beat	15	8	7	(Q) unclassifiable beat	15	8	7
(OTHER) nonbeat annotations	2991	1515	1476	(+) rhythm change	1173	616	557
				(~) change in signal quality	573	289	284
				(!) ventricular flutter wave	472	472	0
				(") comment annotation	437	2	435
				(x) Nonconducted P-wave (blocked APC)	193	58	135
				(|) isolated QRS-like artifact	131	66	65
				([) start of ventricular flutter/fibrillation	6	6	0
				(]) end of ventricular flutter/fibrillation	6	6	0

Since some nonbeat annotations exist in the MIT-BIH database, this article marks them as the OTHER superclass. The relevant data should be excluded from the following process of constructing datasets.

2.2. Overall Design. The overall algorithm design is shown in Figure 2:

(1) The real-time ECG signal sequence is cut into ECG segments with a 200 ms time window

(2) The algorithm detects the QRS complex positions with Classifier 1 in the ECG segments

(3) The algorithm caches ECG data and extracts the time-domain feature (named ECG_RRR) based on the last three QRS complex positions

(4) Classifier 2 predicts arrhythmias using the ECG_RRR feature

The algorithm's output is the arrhythmia type and the corresponding QRS complex position information. As

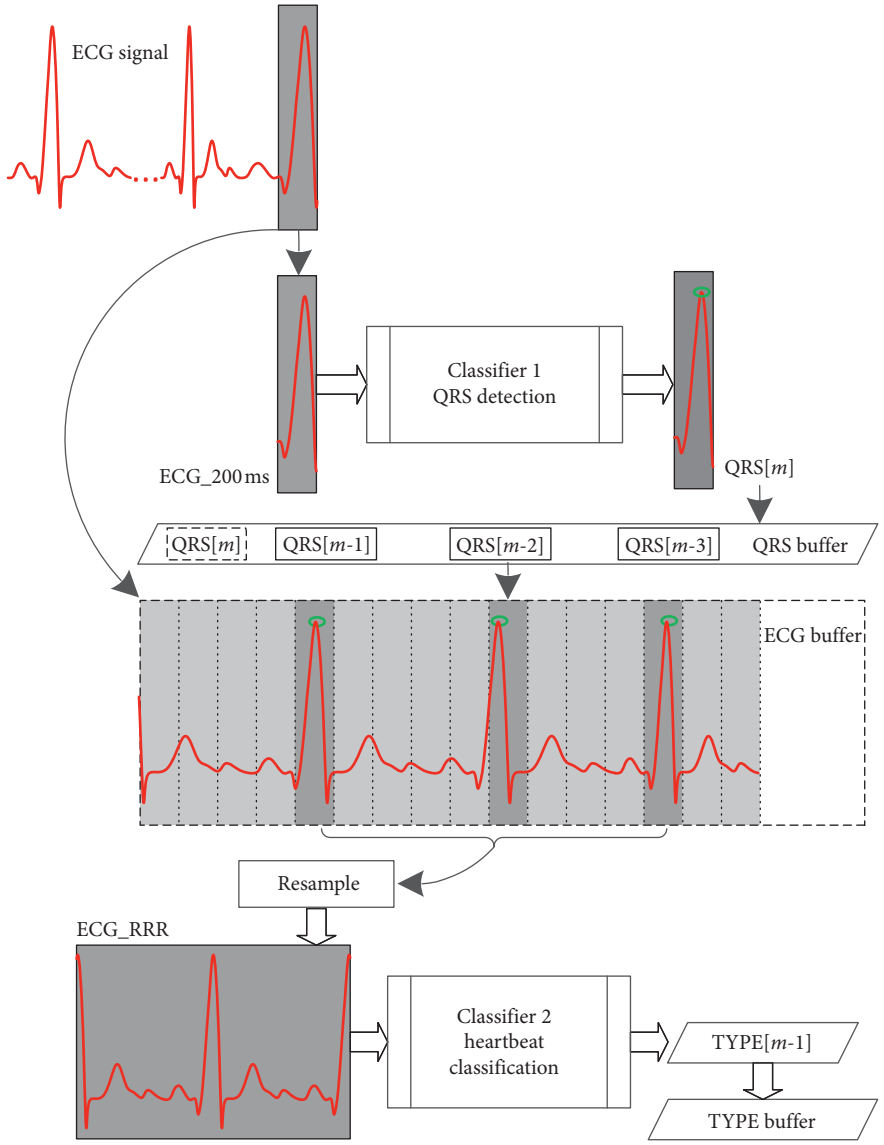

Figure 2: Schematic diagram of the overall algorithm.

shown in Figure 2, the ECG_RRR feature is a resampled ECG sequence with a fixed length. The starting point of the sequence is the last R wave point of the current heartbeat cycle, and the endpoint is the next R wave point of the current heartbeat cycle. Therefore, the ECG_RRR feature contains the complete ECG signal of the current heartbeat cycle. Meanwhile, to improve the accuracy of QRS complex detection, Classifier 1 adopts a step-window voting mechanism to ensure that the moving window contains complete QRS complex information. Since the preprocessing of the ECG signal (such as removing baseline offset, noise, and power-line interference) should be done at the acquisition devices, this algorithm does not consider signal preprocessing.

The 200 ms time window is selected because it not only guarantees real-time performance but also ensures that each two adjacent time windows contain no more than one complete QRS complex.

The real-time ECG signal is cut into ECG segments (named ECG_200 ms, length is 72 when the sample rate is 360 Hz in this paper) by a 200 ms time window, expressed by

$$\text{ECG\_200 ms} = \text{ECGs}[n: n + 72]. \tag{1}$$

The ECG_200 ms is input into Classifier 1, which outputs an integer qrs. $qrs = 0$ means that the input ECG segment does not contain QRS complexes. $0 < qrs \leq 72$ means that the input ECG segment contains QRS complexes and the center position of the QRS complex is qrs. The position of this QRS complex in the ECG signal is $qrs + n$ and is stored in the QRS buffer.

After obtaining the position of the last three consecutive QRS complexes ($QRS[m-2]$, $QRS[m-1]$, and $QRS[m]$), the heart rate (HR) and heart rate variability (HRV) are calculated by

$$HR_{m-1} = \frac{(QRS_{m-1} - QRS_{m-2})}{fs} \times 60,$$
$$HR_m = \frac{(QRS_m - QRS_{m-1})}{fs} \times 60, \quad (2)$$
$$HRV_m = \frac{HR_m - HR_{m-1}}{HR_{m-1}} \times 100\%,$$

where fs is the sampling rate of ECG (360 Hz in this paper).

Only when HR_{m-1} and HR_m are both within the normal range (30~150), the three QRS positions are considered valid. Then, the ECG_RRR feature is obtained by resampling the ECG signal in the ECG buffer between QRS_{m-2} and QRS_m into 360 data points.

Finally, the ECG_RRR feature is input into Classifier 2 which outputs the arrhythmia classification TYPE_{m-1} of the heartbeat cycle at QRS_{m-1}.

The ECG_RRR feature is selected for arrhythmia heartbeat classification because it contains both the complete information of the current heartbeat cycle and partial information of the preceding and succeeding heartbeat cycles as the heart rate variation information.

Classifiers 1 and 2 are both based on deep learning models. According to the characteristics of the input features, Classifier 1 uses a Deep Neural Network (DNN) model of fully connected layers, also called Feedforward Neural Network (FFNN). Classifier 2 uses the Convolutional Neural Network (CNN) model. CNN is a neural network with multiple hidden layers such as fully connected layers, convolutional layers, and pooling layers. CNN is a type of DNN with convolutional layers and pooling layers to extract and abstract features.

For the fully connected layers, which are directly connected layer by layer, the forward propagation from the layer $l-1$ with m features to the layer l with n features is

$$X^{(l)} = f^{(l)}\left(W^{(l)}X^{(l-1)} + B^{(l)}\right). \quad (3)$$

The formula in the matrix form is

$$\begin{bmatrix} x_1^{(l)} \\ \vdots \\ x_n^{(l)} \end{bmatrix} = f^{(l)}\left(\begin{bmatrix} w_1^{(l)1} & \cdots & w_1^{(l)n} \\ \vdots & \ddots & \vdots \\ w_m^{(l)1} & \cdots & w_m^{(l)n} \end{bmatrix}\begin{bmatrix} x_1^{(l-1)} \\ \vdots \\ x_n^{(l-1)} \end{bmatrix} + \begin{bmatrix} b_1^{(l)} \\ \vdots \\ b_n^{(l)} \end{bmatrix}\right), \quad (4)$$

where $X^{(l-1)}$ is the feature vector of the layer $l-1$, a column vector of length m, $X^{(l)}$ is the feature vector of the layer l, a column vector of length n, $W^{(l)}$ is the weight matrix of the layer l, a matrix of m rows and n columns, $B^{(l)}$ is the bias vector of the layer l, a column vector of length n, and $f^{(l)}$ is the nonlinear activation function of the layer l.

Each convolution layer contains several convolution kernels. The features of the previous layer are convolved with the corresponding convolution kernels to output new features. For the layer $l-1$ with depth i (the feature maps are $X_1^{(l-1)}, \ldots, X_i^{(l-1)}$), after the operation of the convolution layer (with learnable convolution kernels $K_{1,1}^{(l)}, \ldots, K_{i,1}^{(l)}, \ldots, K_{1,j}^{(l)}, \ldots, K_{i,j}^{(l)}$ and learnable bias $B_1^{(l)}, \ldots, B_j^{(l)}$), the layer l with depth j (the feature maps are $X_1^{(l)}, \ldots, X_j^{(l)}$) is obtained. The calculation formula of each feature map is

$$X_j^{(l)} = f^{(l)}\left(\sum_{i \in M_j}\left(X_i^{(l-1)} \otimes K_{i,j}^{(l)}\right) + B_j^{(l)}\right), \quad (5)$$

where M_j is the set of input feature maps, $\otimes$ is the convolution operation, and $f^{(l)}$ is the activation function.

For a CNN model, there is usually a pooling layer following the convolutional layer to reduce the feature size. After several convolutional layers and pooling layers, a flatten layer transforms the features into a vector, and then, the output is obtained through several fully connected layers.

In this study, the hidden layer uses the linear rectification function (Rectified Linear Unit, ReLU) as the activation function, which has the advantage of fast convergence. For an input column vector $[x_1, \ldots, x_n]^{\mathrm{T}}$, the function output is

$$f(\mathbf{x})_i = \begin{cases} x_i, & x_i > 0, \\ 0, & x_i \le 0, \end{cases} \quad \text{for } i \text{ in } \{1, \ldots, n\}, \mathbf{x} = \left[x_1, \ldots, x_n\right]^T. \quad (6)$$

In this paper, the output layer uses the SoftMax activation function to normalize the output into a probability distribution. For an input column vector $[x_1, \ldots, x_n]^{\mathrm{T}}$, the function output is

$$f(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}}, \quad \text{for } i \text{ in } \{1, \ldots, n\}, \quad \mathbf{x} = \left[x_1, \ldots, x_n\right]^T. \quad (7)$$

The loss function evaluates the difference between the predicted output $\hat{\mathbf{y}}$ and the true value $\mathbf{y}$. In this paper, the cross-entropy loss function is used, and its function for an n-class classification problem is

$$\text{Loss}(\hat{y}, \mathbf{y}) = -\sum_{i=1}^{n}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right], \quad (8)$$

where $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_n]$ is the predicted output and $\mathbf{y} = [y_1, \ldots, y_n]$ is usually obtained by the one-hot encoding of the true class.

In this study, the Adam algorithm optimizes the model parameters. It has the advantages of the adaptive learning rate, fast convergence, and stable results. This article uses TensorFlow 2, a deep learning open-source tool released by Google, to build, train, and test the models.

2.3. Dataset Construction. According to the overall algorithm design, there are two classifiers, and the cores of them are deep learning-based neural network models. They are trained on two datasets, named ECG_200 ms_POS_72 and ECG_RRR_TYPE_360, respectively.

The generation of the ECG_200 ms_POS_72 and ECG_RRR_TYPE_360 datasets from the MIT-BIH database is shown in Figure 3. To ensure the validity of the dataset, we should eliminate the relevant data in the superclass OTHER.

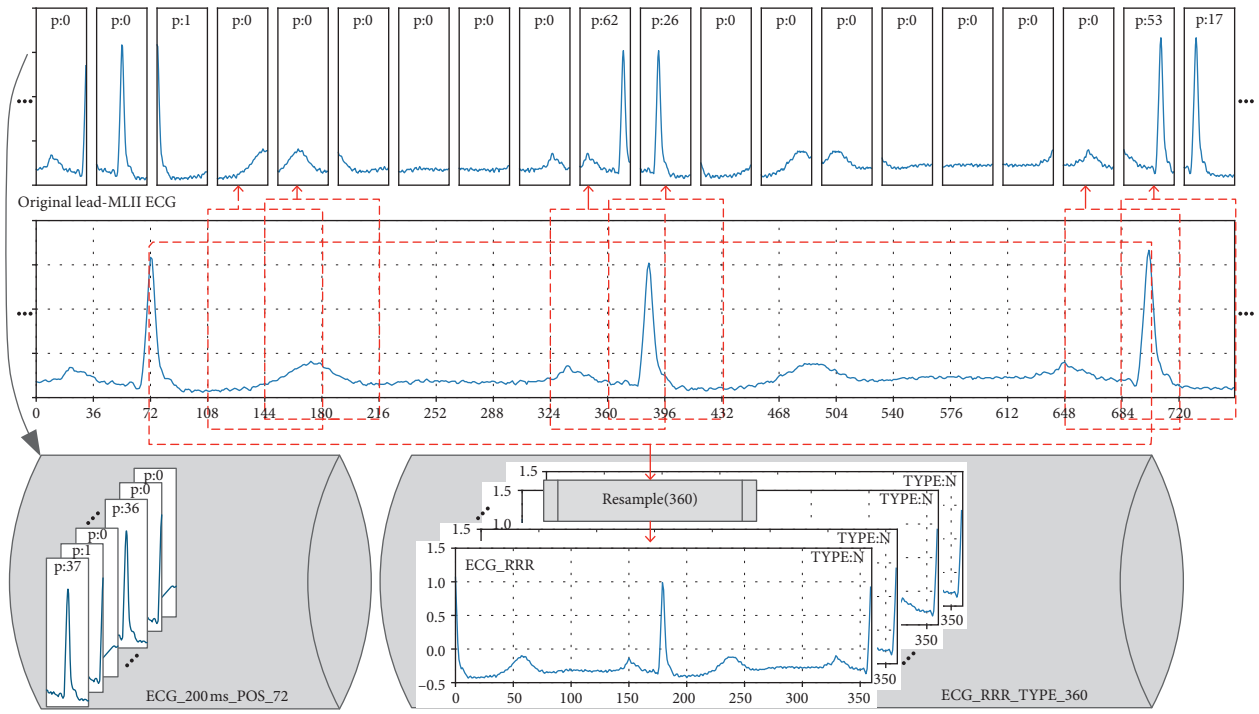

Figure 3: Schematic diagram of generating the ECG_200 ms_POS_72 and ECG_RRR_TYPE_360 datasets from the MIT-BIH database. The ECG_200 ms_POS_72 dataset is used to train Classifier 1 for QRS complex position prediction. Each of these samples contains an ECG segment (ECG_200 ms) and the corresponding QRS complex position label. The ECG_200 ms is a 200 ms ECG-MLII lead ECG segment with a length of 72 (when the sampling rate is 360 Hz) in mV. The label p is the center position of the QRS complex (an integer in [0, 71], where 0 means no QRS complex). The ECG_RRR_TYPE_360 dataset is used to train Classifier 2 for arrhythmia classification. Each of these samples contains an ECG_RRR feature and a corresponding arrhythmia type (TYPE). An ECG_RRR ECG segment is the MLII lead ECG data between every three QRS complexes resampled to a fixed length of 360 in mV. The data label TYPE is the arrhythmia type corresponding to the middle QRS complex (an integer in [0, 12], which represents 13 classes of arrhythmia).

Meanwhile, when intercepting ECG_RRR, a moving window with a length of 200 ms (72) and a step of 100 ms (36) splits the original data.

Figure 4 shows partial waveform diagrams of various ECG_RRR features. There are apparent differences in the morphological characteristics of various ECG_RRR features, which further proves the effectiveness of ECG_RRR features for arrhythmia classification.

This paper constructs the two datasets (ECG_200 ms_POS_72 and ECG_RRR_TYPE_360) after processing all 44 records. Table 2 shows the description of the constructed two datasets. Table 3 shows the distribution of samples in the ECG_200 ms_POS_72 dataset, which contains a total of 794,376 samples (sample_A). Among them, 590,230 samples (sample_N) are negative samples with label 0, indicating that no QRS complex is included. The other 204,146 samples (sample_P) are positive examples with positive integer labels, meaning that the QRS complex is included. The positive label values are approximately uniformly distributed in [1, 71]. The average value (P_ave) of the distribution is 2875.30, the maximum value (P_max) is 3019, the minimum value (P_min) is 2784, and the variance (P_s) is 55.13. Table 4 shows the distribution of the samples in the ECG_RRR_TYPE_360 dataset, which contains a total of 97,898 records. Although the data of Categories F (fusion beat) and Q (unknown beat) are imbalanced (the amount of data is small), these two categories are not the main classification targets of the classifier.

2.4. QRS Position Detection. The QRS position detection subalgorithm marks the position of the QRS complex in the ECG signal so that the overall algorithm can segment the heartbeat. In this paper, the input of the QRS position detection subalgorithm is the 200 ms ECG segment with a length of 72, and the output is the QRS complex position (an integer in [1, 71]) or 0 (when the QRS complex is not included).

This subalgorithm consists of a 72-class FFNN-based model (named Model_QRS) and a QRS detection strategy.

The Model_QRS model is trained and evaluated on the DS1 group of the ECG_200 ms_POS_72 dataset, which is randomly divided into a training set (train set) and a validation set (valid set) with a ratio of 8 : 2. The test set consists of the data in the DS2 group of the ECG_200 ms_POS_72 dataset.

To reduce the computational cost and meet the requirements of edge computing equipment, it is a good idea to use as small a model as possible to obtain better results. Therefore, grid searching is employed to identify the optimal model for QRS position detection. This paper designs all hidden layers to the same size to facilitate searching. As

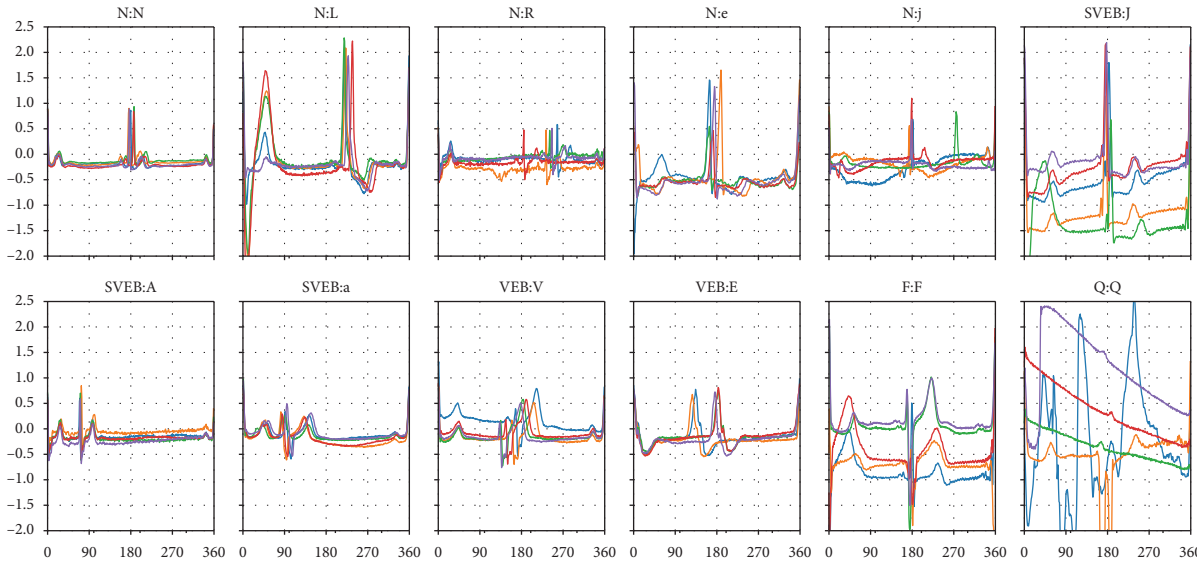

Figure 4: Partial waveform diagrams of various ECG_RRR features.

Table 2: The description of the constructed two datasets.

Dataset	ECG_200 ms_POS_72	ECG_RRR_TYPE_360
Input	200 ms ECG segment (mV)	"ECG_RRR" feature (mV)
Input shape	72	360
Output	QRS complex position ([0, 71])	Arrhythmia type ([0, 12])
Output shape	1	1
Total samples	794376	97898

Table 3: Statistics of label distribution in the ECG_200 ms_POS_72 dataset.

Group	sample_A	sample_N	sample_P	P_ave	P_max	P_min	P_s
DS1	397188	293797	103391	1456.21	1551	1384	36.39
DS2	397188	296433	100755	1419.08	1490	1318	36.88
DS1 + DS2	794376	590230	204146	2875.30	3019	2784	55.13

Table 4: Distribution statistics of samples in the ECG_RRR_TYPE_360 dataset.

Superclass	DS1	DS2	Subclass	DS1	DS2
N	45055	43237	N	37310	35503
			L	3941	4122
			R	3774	3400
			e	16	0
			j	14	212
SVEB	576	1741	A	466	1678
			a	76	35
			J	32	28
			S	2	0
VEB	3412	3104	V	3309	3103
			E	103	1
F	373	385	F	373	385
Q	8	7	Q	8	7

shown in Figure 5, the number of hidden layers varies from 1 to 8. Since the input feature size is 72, the candidate size of each hidden layer is {9, 18, 36, 72, 144}.

Sometimes there is a slight error between the marked points in the MIT-BIH database and the actual QRS complex position. It also allows a certain error to exist in the QRS

	Size = 9	Size = 18	...	Size = 144
Layers = 1	DNN (9, 1)	DNN (18, 1)	...	DNN (144, 1)
Layers = 2	DNN (9, 2)	DNN (18, 2)	...	DNN (144, 2)
Layers = 3	DNN (9, 3)	DNN (18, 3)	...	DNN (144, 3)
...	...	...	...	...
Layers = 8	DNN (9, 8)	DNN (18, 8)	...	DNN (144, 8)

Figure 5: The search grid for an appropriate FFNN model for QRS detection.

complex position prediction in practice. Therefore, this article concerns more about the accuracy (named acc2) that the QRS complex position is within the tolerated error range (±5).

The sizes of the input and output layers are both 72 (the result of the QRS complex position is the index of the maximum value in the output layer). After training the model with the cross-entropy loss, Adam optimizer, and early stopping strategy (if the loss of the model is less than $1e-5$ for consecutive five epochs, stop training to prevent overfitting), we get the result of grid searching, which is shown in Figure 6. As can be seen from the figure, with the increase of the number and size of hidden layers, the accuracy of the model improves, but when the size of each hidden layer is greater than 36 and the number of hidden layers is greater than three, the improvement of acc2 is not significant. Therefore, in this paper, an FFNN model with three hidden layers and 36 neurons in each hidden layer is selected as the model for Classifier 1 for QRS position detection. Figure 7 shows the structure of the final selected FFNN model. Table 5 shows the details of each layer of the model. The number of parameters in this model is 7956, the model size is about 32 kB, and the amount of computation required for the model is 16092 FLOPs (floating-point operations [39]). Figure 8 shows the accuracy and loss curves of the model training process.

Finally, the acc2 on the test set is 96.8%.

The algorithm uses a strategy based on a step-window voting mechanism when the model is deployed in production to improve the accuracy, as shown in Figure 9:

(1) The algorithm caches the latest ECG segment with a length of 144 and a duration of 400 ms and uniformly uses N windows with length 72 to obtain N 200 ms ECG segments (ecg_w[72][N]).

(2) The algorithm inputs the (ecg_w[72][N]) to the FFNN model to get the results of N QRS complex positions (QRSs[N]).

(3) The final QRS position (qrs) and total votes (vote) are calculated according to QRSs[N]. The prediction result (qrs) of this QRS complex is retained only when (vote) is greater than threshold V and the position from the previous QRS complex is greater than threshold L. The values of thresholds V and L and the number of windows N can be adjusted according to requirements.

In this paper, we choose $N = 8, V = 4$, and $L = 72$ and use all 22 records of the DS2 group in the original data to evaluate Classifier 1 (for QRS position detection). The recall rate, precision rate, and overall accuracy of QRS complex position prediction are 98.0%, 99.5%, and 97.6%, respectively.

2.5. Heartbeat Classification. The heartbeat classification subalgorithm uses the ECG_RRR feature. Since there are only 13 classes of arrhythmia in MIT-BIH, this subalgorithm is a 13-class arrhythmia classification model that can get 5-class arrhythmia classification according to the class hierarchy.

Since ECG_RRR features have relatively strong morphological characteristics, the core of this subalgorithm is a 13-class classifier based on the CNN model (named Model_TYRE) and uses the ECG_RRR_TYPE_360 dataset for training and evaluation. We use the DS1 group of the ECG_RRR_TYPE_360 dataset as training data, which is further randomly split into a training set (train set) and a validation set (valid set) with a ratio of 8 : 2. The test data comes from the DS2 group of the ECG_RRR_TYPE_360 dataset.

The input of the model is the ECG_RRR feature proposed in this paper, with a length of 360 and clear morphological characteristics. Therefore, a one-dimensional CNN model constructs the model, and the result shows that this model has a good performance.

After considering the tradeoff between the size and accuracy of the model, we select a structure of the CNN-based model and show it in Figure 10. Table 6 shows the details of the model. The number of parameters in this model is 6273, the model size is about 25 kB, and the amount of computation required is 105678 FLOPs (floating-point operations [39]). The input of the model is the ECG_RRR feature with a depth of 1 and a length of 360. The model then uses a cascade of three one-dimensional convolutional layers and pooling layers to transform the data into features with depth 20 and length 6. Then, a flatten layer flattens the features to a vector of length 120. Finally, two full connection layers of size 30 and one output layer obtain an output of length 13, each representing the probability of one of the 13 classes. The output with the highest probability is the model's predicted class, and then, the 5-class prediction is obtained according to the class hierarchy.

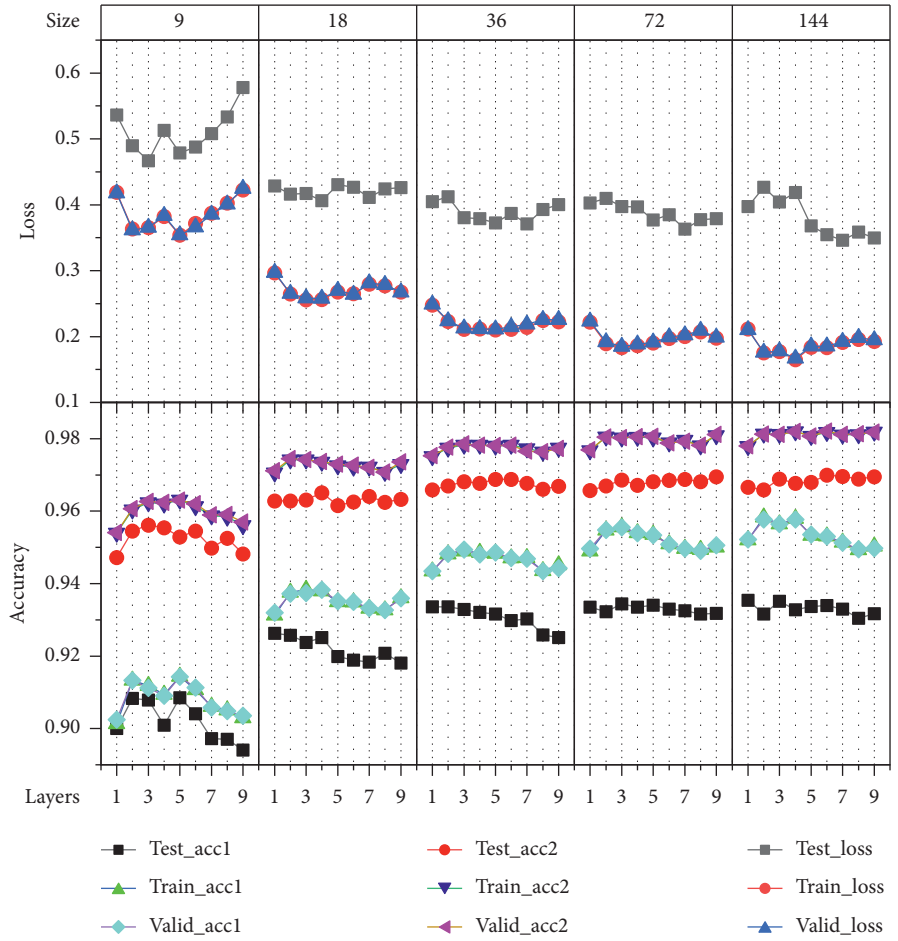

Figure 6: Training results of the models with different layer numbers and sizes in grid search.

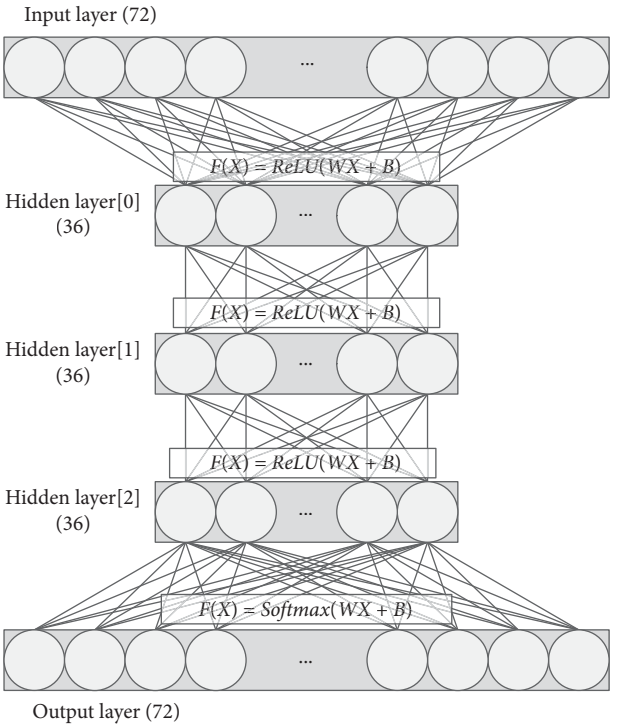

Figure 7: The structure of the final selected FFNN model for QRS complex position detection.

Table 5: Layers of the final selected FFNN model for QRS complex position detection.

Index	Name	Layer type	Active function	Output shape	Params
0	Input	Input	—	72	0
1	Hidden0	Dense	ReLU	36	2628
2	Hidden1	Dense	ReLU	36	1332
3	Hidden2	Dense	ReLU	36	1332
4	Output	Output	SoftMax	72	2664

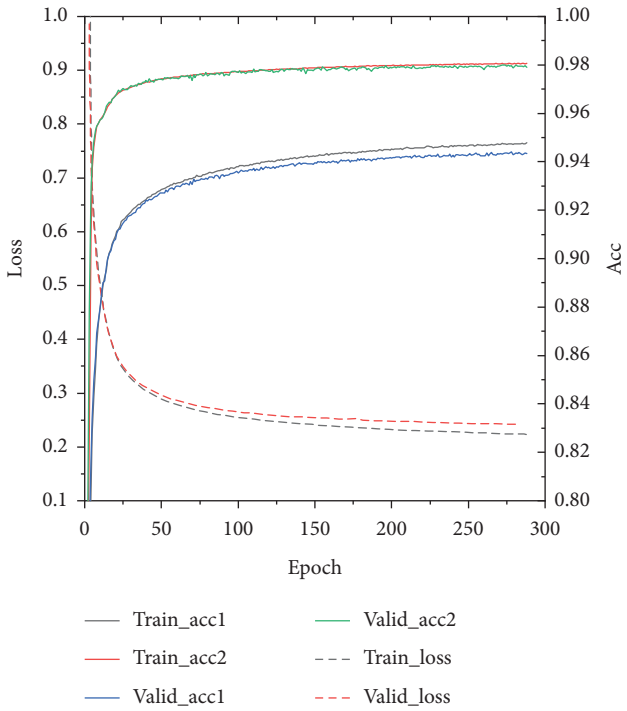

Figure 8: The accuracy and loss curves during the training process of the FFNN model for QRS position prediction.

To prevent overfitting, dropout is applied on pooling layers and full connection layers during model training and randomly disables 30% of neuron connections.

After training the model with the cross entropy, Adam optimizer, and early stopping strategy (if the loss of the model is less than $1e-5$ for consecutive five epochs, it stops training to prevent overfitting), the accuracy and loss curves during the training process are shown in Figure 11. Since the dropout is applied in the training process, the training loss is higher than the validation loss, and the training accuracy is lower than the validation accuracy.

Since the data in the DS1 and DS2 groups belong to different patients and the model only learns on 80% of the data in the DS1 group, the performance of the model on the DS2 group is an indicator of interpatient performance. The accuracy of 13-class classification is not good enough (only 77.0%), but the accuracy of 5-class classification according to the hierarchy is as high as 94.2%.

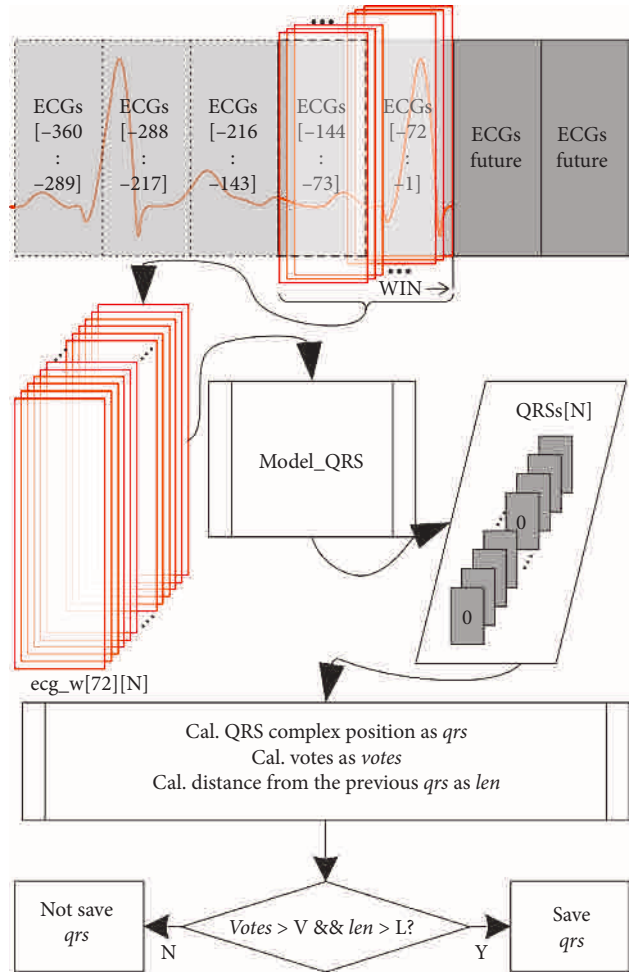

Figure 9: QRS detection strategy based on a step-window voting mechanism.

3. Results and Discussion

In the experimental test, the algorithm first uses Classifier 1 to predict the QRS complex position, which assists in segmenting the ECG_RRR feature. It then uses Classifier 2 to predict the corresponding arrhythmia type. Finally, the QRS position predictions and the corresponding arrhythmia predictions are compared with the ground truth in the database to obtain the evaluation results.

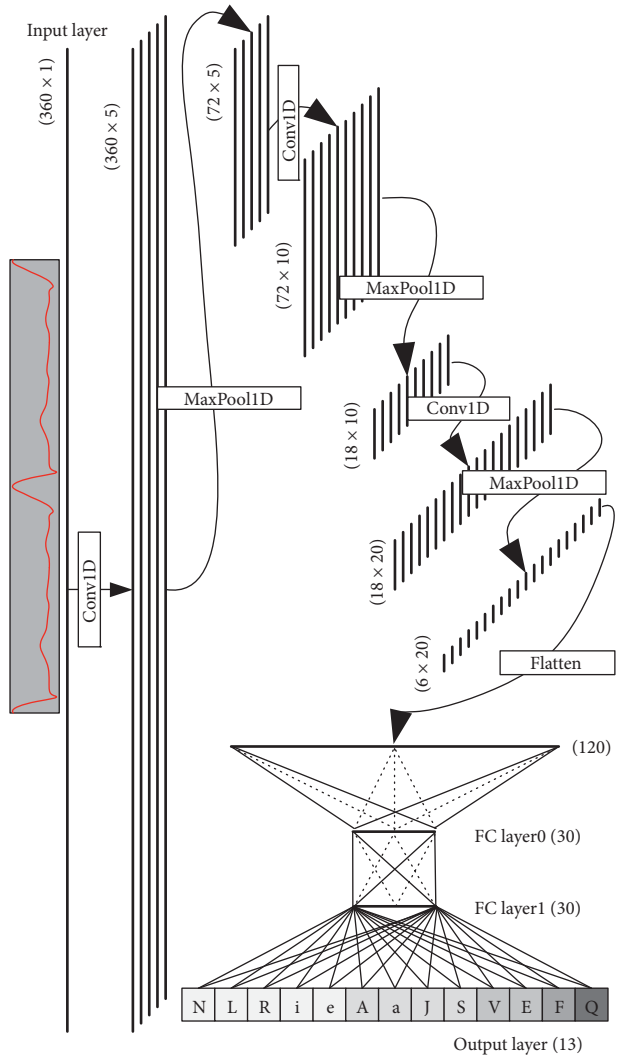

Figure 10: The structure of the CNN-based heartbeat classification model.

Table 6: Layers of the CNN-based heartbeat classification model.

Index	Name	Layer type	$F(x)$	Shape	Params
0	Input	Reshape	—	360, 1	0
1	Hidden1	Conv1D	ReLU	360, 5	30
2	Hidden2	MaxPool1D	ReLU	72, 5	0
3	Hidden3	Conv1D	ReLU	72, 10	260
4	Hidden4	MaxPool1D	ReLU	18, 10	0
5	Hidden5	Conv1D	ReLU	18, 20	1020
6	Hidden6	MaxPool1D	ReLU	6, 20	0
7	Hidden7	Flatten	ReLU	120	0
8	Hidden8	Dense	ReLU	30	3630
9	Hidden9	Dense	ReLU	30	930
10	Output	Dense	SoftMax	13	403

This paper uses all the 44 patients' MLII lead original ECG signals in Groups DS1 and DS2 of the MIT-BIH arrhythmia database to test the overall algorithm. Besides, the interpatient performance of the algorithm is obtained by testing on the data in the DS2 group, which is also an important evaluation metric in practice.

For each of these records, the classification pipeline is as follows: firstly, the ECG signal of MLII lead is sliced by a 200 ms time window and passed into the algorithm chronologically. The algorithm then predicts the position of the QRS complex and the corresponding arrhythmia class in real time. After all data in this record is entered, the predictions

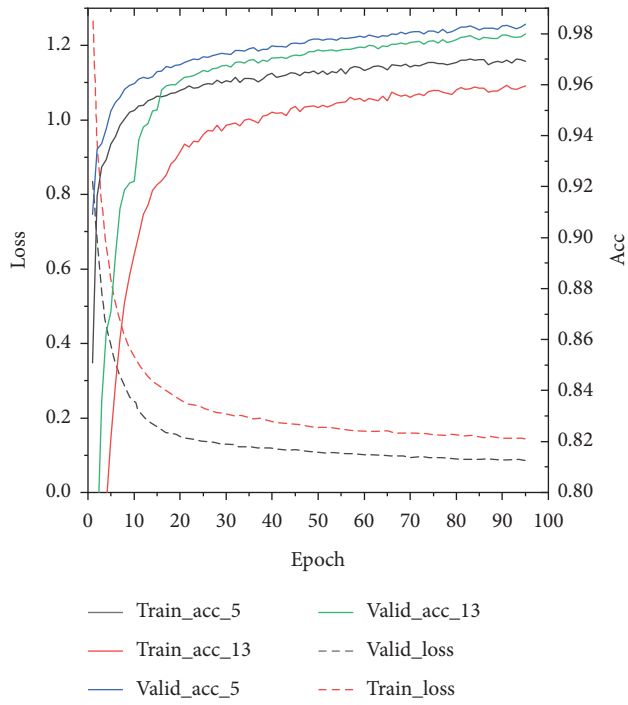

Figure 11: The accuracy and loss curves during the training process of the heartbeat classification model based on CNN.

of the QRS complex position sequence (QRS_pred) and the corresponding heartbeat type sequence (TYPE_pred) are obtained. Meanwhile, the ground truth QRS complex position sequence (QRS_true) and the corresponding heartbeat type sequence (TYPE_true) are obtained by analyzing the marker information in this record. Finally, by comparing QRS_pred and TYPE_pred with QRS_true and TYPE_true, respectively, the corresponding confusion matrix and statistical matrix are obtained. When comparing QRS_true with QRS_Pred, a certain error of ±5 is allowed.

The recall rate (R), precision rate (P), and overall accuracy (Acc) are the evaluation metrics of classification performance. For an n-class classification problem with classes $x_1, x_2, \ldots, x_n$, the calculation formulas of precision rate, recall rate, and overall accuracy are as follows:

$$P_{x_i} = \frac{N_{x_i}^{x_i}}{N^{x_i}}, \tag{9}$$

$$R_{x_i} = \frac{N_{x_i}^{x_i}}{N_{x_i}}, \tag{10}$$

$$\text{Acc} = \frac{\sum_1^n N_{x_i}^{x_i}}{\sum_1^n N_{x_i}} = \frac{\sum_1^n N_{x_i}^{x_i}}{\sum_1^n N^{x_i}}, \tag{11}$$

where $N_{x_i}^{x_j}$ represents the number of samples whose true class is x_i and predicted class is x_j, $N_{x_i}^{x_i}$ represents the number of samples whose true class and predicted class is both x_i, N_{x_i} represents the number of samples whose true class is x_i, and N^{x_j} represents the number of samples whose predicted class is x_j.

For the real-time ECG arrhythmia classification in this paper, there is a specific case in which the prediction of the QRS complex position is incorrect. This article marks this specific case as 0, and the quantity statistics are represented by $N_0^{x_i}$ and $N_{x_i}^0$, where $N_0^{x_i}$ represents the number of samples whose predicted class is x_i in false-positive samples and $N_{x_i}^0$ represents the number of samples whose true class is x_i in false-negative examples.

For the 5-class arrhythmia classification, the results are recorded in the confusion matrix as Table 7 and the statistical table as Table 8.

The recall rate, precision rate, and overall accuracy of N, S (SVEB), V (VEB), F, and Q are calculated according to the 6-class case regarding equations (9)–(11). Specifically, the numbers of false-positive (FP), false-negative (FN), and true-positive (TP) samples of QRS position prediction are calculated by

$$\begin{aligned} \text{FP}_{\text{QRS}} &= N_0, \\ \text{FN}_{\text{QRS}} &= N^0, \\ \text{TP}_{\text{QRS}} &= N^N + N^S + N^V + N^F + N^Q - N_0. \end{aligned} \tag{12}$$

The precision rate (P), recall rate (R), and overall accuracy (Acc) of QRS complex position prediction are calculated by

Table 7: Example of a confusion matrix for the recording of 5-class arrhythmia classification test results.

		Predicted						
		N	S	V	F	Q	0	Sum
True	N	N_N^N	N_N^S	N_N^V	N_N^F	N_N^Q	N_N^0	N_N
	S	N_S^N	N_S^S	N_S^V	N_S^F	N_S^Q	N_S^0	N_S
	V	N_V^N	N_V^S	N_V^V	N_V^F	N_V^Q	N_V^0	N_V
	F	N_F^N	N_F^S	N_F^V	N_F^F	N_F^Q	N_F^0	N_F
	Q	N_Q^N	N_Q^S	N_Q^V	N_Q^F	N_Q^Q	N_Q^0	N_Q
	0	N_0^N	N_0^S	N_0^V	N_0^F	N_0^Q	–	N_0
	Sum	N^N	N^S	N^V	N^F	N^Q	N^0	SUM

Table 8: Example of a statistical table of 5-class arrhythmia classification test results.

No.		N	S	V	F	Q	Acc
XX	(P)	P^N	P^S	P^V	P^F	P^Q	Acc
	(R)	R_N	R_S	R_V	R_F	R_Q	

Table 9: Confusion matrix of 5-class arrhythmia classification test results on DS1.

		Predicted						
		N	S	V	F	Q	0	Sum
True	N	45141	74	221	10	0	420	45866
	S	445	418	8	1	0	72	944
	V	209	83	3151	16	0	329	3788
	F	84	0	37	288	0	6	415
	Q	2	0	0	0	0	6	8
	0	121	3	67	0	0	—	191
	Sum	46002	578	3484	315	0	833	51212

Table 10: Confusion matrix of 5-class arrhythmia classification of test results on DS2.

		Predicted						
		N	S	V	F	Q	0	Sum
True	N	42563	446	411	9	0	830	44259
	S	1554	190	47	0	0	46	1837
	V	172	21	2929	9	0	90	3221
	F	273	0	106	1	0	8	388
	Q	1	0	3	0	0	3	7
	0	115	1	116	1	0	—	233
	Sum	44678	658	3612	20	0	977	49945

$$P_{\mathrm{QRS}} = \frac{\mathrm{TP}_{\mathrm{QRS}}}{\mathrm{TP}_{\mathrm{QRS}} + \mathrm{FP}_{\mathrm{QRS}}},$$

$$R_{\mathrm{QRS}} = \frac{\mathrm{TP}_{\mathrm{QRS}}}{\mathrm{TP}_{\mathrm{QRS}} + \mathrm{FN}_{\mathrm{QRS}}}, \tag{13}$$

$$\mathrm{Acc}_{\mathrm{QRS}} = \frac{\mathrm{TP}_{\mathrm{QRS}}}{\mathrm{TP}_{\mathrm{QRS}} + \mathrm{FN}_{\mathrm{QRS}} + \mathrm{FP}_{\mathrm{QRS}}}.$$

After testing all records in the database, 44 confusion matrices are obtained. According to the DS1 and DS2 groups, two result Tables 9 and 10 and a statistics Table 11 are obtained after synthesizing them.

From Tables 9 and 10, it can be calculated that, for the DS1 group, the recall rate, precision rate, and overall accuracy are 98.3%, 99.6%, and 98.0%, respectively. For the DS2 group, they are 98.0%, 99.5%, and 97.6%, respectively.

Table 11 shows that, on the data in the MIT-BIH database, the overall accuracy for 5-class arrhythmia classification is 93.6%, and the interpatient accuracy is 91.5%. The statistics of each patient's 5-class arrhythmia classification results in Group DS2 are shown in Table 12, and the distribution of overall accuracy is shown in Figure 12. As shown in the table, except for the extremely low accuracy of Patient 232 and the low accuracy of Patients 222, 219, and 213, the accuracies of the other 18 patients are all above 90%.

Table 11: Statistics of 5-class arrhythmia classification test results.

No.		N	S	V	F	Q	Acc
DS1	(P)	0.981	0.723	0.904	0.914	—	0.957
	(R)	0.984	0.443	0.832	0.694	0.000	
DS2	(P)	0.953	0.289	0.811	0.050	—	0.915
	(R)	0.962	0.103	0.909	0.003	0.000	
ALL	(P)	0.967	0.492	0.857	0.863	—	0.936
	(R)	0.973	0.219	0.867	0.360	0.000	

Table 12: Statistics of 5 arrhythmia classes for each patient on DS2.

No.		N	S	V	F	Q	Acc
103	(R)	0.999	1.000	—	—	—	0.999
	(P)	1.000	0.500	—	—	—	
100	(R)	0.999	0.971	1.000	—	—	0.998
	(P)	0.999	1.000	1.000	—	—	
212	(P)	0.998	—	—	—	—	0.996
	(R)	0.998	0.000	0.000	—	—	
...	...	...	...	...	...	...	...
219	(P)	0.780	0.857	0.906	0.000	—	0.782
	(R)	0.998	0.019	0.287	—	—	
222	(P)	0.798	0.306	—	—	—	0.756
	(R)	0.938	0.780	0.000	—	—	
232	(R)	0.228	0.909	0.000	—	—	0.247
	(P)	0.980	0.036	—	—	—	

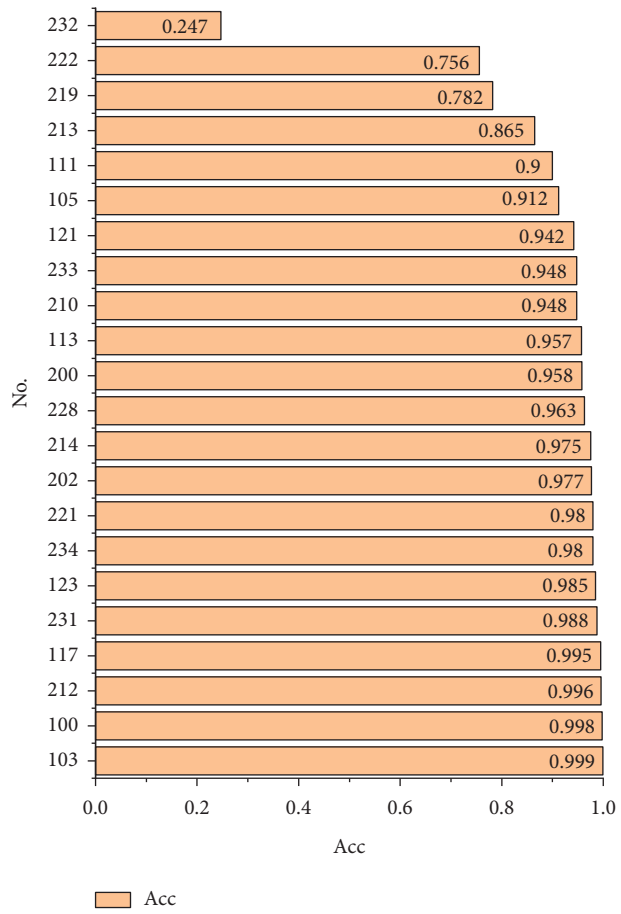

Figure 12: The overall accuracy of each patient's 5-class arrhythmia classification on DS2.

Table 13: Comparison of the proposed algorithm with other studies.

Method	Acc	P (N)	R (N)	P (SVEB)	R (SVEB)	P (VEB)	R (VEB)	P (F)	R (F)
Proposed	0.915	0.953	0.962	0.289	0.103	0.811	0.909	0.050	0.003
Proposed*	0.942	0.983	0.958	0.123	0.333	0.956	0.862	0.008	0.176
Luo et al. [26]	0.893	0.930	0.953	0.473	0.154	0.668	0.604	0.200	0.500
Mar et al. [14]	0.890	0.992	0.942	0.567	0.862	0.934	0.924	0.177	0.664
Alvarado et al. [27]	0.936	0.992	0.942	0.567	0.862	0.934	0.924	0.177	0.664
Ye et al. [28]	0.882	0.982	0.900	0.551	0.564	0.603	0.847	0.058	0.358
Zhang et al. [29]	0.883	0.990	0.889	0.360	0.791	0.928	0.855	0.137	0.938
Niu et al. [30]	0.923	0.974	0.939	0.732	0.766	0.578	0.851	0.449	0.384

Table 13 summarizes the comparisons of the proposed algorithm and state-of-the-art methods which do not require expert intervention. It demonstrates that the proposed algorithm has an advantage in overall accuracy, although the recall rate and precision rate of the SVEB classification are lower than those of the other studies. The algorithm of this paper uses the heartbeat segmentation provided by Classifier 1 for arrhythmia classification, and its overall accuracy may be affected by the error of heartbeat segmentation. However, many other studies just segment heartbeat by the marked points in the database for arrhythmia classification. For a fair comparison, the results of Classifier 2 on the DS2 group in the ECG_RRR_TYPE_360 dataset are also listed (proposed $*$ rows in the table), which is the interpatient arrhythmia classification performance using the heartbeat markers in the database.

4. Conclusions

Real-time monitoring of ECG and intelligent diagnosis in daily life are of great significance to reduce the risk of cardiovascular disease. With the development of wearable ECG measurement technology and edge computing, a real-time arrhythmia diagnosis system combining the two is a solution in which a reliable real-time arrhythmia classification algorithm is the core. This paper introduces a novel arrhythmia classification model which performs with high accuracy and in real time. The model takes the raw ECG signal as input and segments it with a time window. Then, it detects the QRS complex positions with an FFNN-based model and extracts the time-domain feature (ECG_RRR) for another CNN-based model to predict the arrhythmia type. Experimental results show that our model performs very well on the MIT-BIH dataset. In addition, our model requires low computing power for real-time prediction, which is available on most desktop and mobile processors.

4.1. Practicality. The input of the algorithm is the original ECG signal without complex feature extraction. Moreover, to adapt to different sampling rates, a sampling rate adaptation layer can be added before the algorithm is in practice.

The core of the algorithm is deep learning that is already very mature. There are many excellent deep learning frameworks for cloud computing or mobile devices, such as TensorFlow, Caffe, PyTorch, and MXNet, so the proposed algorithm can be easily deployed.

4.2. Real Time. The cores of the algorithm are two deep learning-based classifiers with simple structure, high performance, and low computational cost.

The input of the algorithm is a 200 ms ECG segment. The latency of detecting the QRS complex position is 400 ms, and the latency of arrhythmia classification is just one heartbeat cycle. The numbers of parameters in the two models are 7956 and 6273, respectively, and the computational requirements are 16092 FLOPs and 105678 FLOPs, respectively. Considering the use of a step-window voting strategy to detect QRS complexes, the comprehensive computation is 0.235 MFLOPs. The device only needs 1.2 MFLOPS (FLOPS means Floating-Point Operations Per Second) of floating-point operation capability to meet the computing needs. At present, desktop-level CPUs and GPUs can reach the magnitude of GFLOPS or even TFLOPS, and mobile processors can reach hundreds of MFLOPS. Therefore, the computational requirement of the algorithm is not a problem.

4.3. Effectiveness. The proposed algorithm includes two deep learning-based classifiers for QRS complex position detection and arrhythmia classification, respectively. Meanwhile, a strategy based on a step-by-step window voting mechanism is proposed to improve QRS complex position prediction accuracy.

For interpatient performance, the recall rate, precision rate, and overall accuracy of the algorithm's interpatient QRS complex position prediction are 98.0%, 99.5%, and 97.6%, respectively. The algorithm has overall accuracies of 91.5% and 75.6% for 5-class and 13-class arrhythmia classification, respectively.

Data Availability

The two datasets for training and evaluating the models constructed in this study are available from the corresponding author upon request and the origin real-world ECG data can be obtained from MIT-BIH database (https://www.physionet.org/content/mitdb/1.0.0/).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Science and Technology Development Project of Jilin Province, China (20190303043SF), Science and Technology Development Project of Jilin Province, China (20200404205YY), and National Key Research and Development Program of China (2018YFF0300806-1).

References

[1] WHO, *WHO|Cardiovascular Diseases (CVDs)*, WHO, Geneva, Switzerlandhttps://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-%20(cvds.

[2] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Generation Computer Systems*, vol. 86, no. 1, pp. 446–455, 2018.

[3] E. J. D. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, "ECG-based heartbeat classification for arrhythmia detection: a survey," *Computer Methods and Programs in Biomedicine*, vol. 127, no. 1, pp. 144–164, 2016.

[4] S. Yu and K. Chou, "Integration of independent component analysis and neural networks for ECG beat classification," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2841–2846, 2008.

[5] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[6] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen, and L. Shen, "ECG beat detection using filter banks," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 192–202, 1999.

[7] Y.-C. Yeh and W.-J. Wang, "QRS complexes detection for ECG signal: the Difference Operation Method," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 3, pp. 245–254, 2008.

[8] O. Sayadi and M. B. Shamsollahi, "A model-based Bayesian framework for ECG beat segmentation," *Physiological Measurement*, vol. 30, no. 3, pp. 335–352, 2009.

[9] J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, "A wavelet-based ECG delineator: evaluation on standard databases," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 4, pp. 570–581, 2004.

[10] R. Poli, S. Cagnoni, and G. Valli, "Genetic design of optimum linear and nonlinear QRS detectors," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 11, pp. 1137–1141, 1995.

[11] Y. H. Hu, W. J. Tompkins, J. L. Urrusti et al., "Applications of artificial neural networks for ECG signal detection and classification," *Journal of Electrocardiology*, vol. 26, no. 1, pp. 66–73, 1993.

[12] K. S. Park, B. H. Cho, D. Lee et al., "Hierarchical support vector machine based heartbeat classification using higher order statistics and hermite basis function," in *Proceedings of the 2008 Computers in Cardiology*, pp. 229–232, Bologna, Italy, September 2008.

[13] G. D. Lannoy, D. Franois, J. Delbeke et al., "Weighted SVMs and feature relevance assessment in supervised heart beat classification,".

[14] T. Mar, S. Zaunseder, J. P. Martínez, M. Llamedo, and R. Poll, "Optimization of ECG classification by means of feature selection," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 8, pp. 2168–2177, 2011.

[15] İ. Güler and E. D. Übeyli, "ECG beat classifier designed by combined neural network model," *Pattern Recognition*, vol. 38, no. 2, pp. 199–208, 2005.

[16] P. d. Chazal, M. O' Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.

[17] M. A. Escalona-Morán, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Electrocardiogram classification using reservoir computing with logistic regression," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 892–898, 2015.

[18] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[19] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "A novel application of deep learning for single-lead ECG classification," *Computers in Biology and Medicine*, vol. 99, no. 1, pp. 53–62, 2018.

[20] P. Paweł and U. R. Acharya, "Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals," *Neural Computing and Applications*, vol. 32, no. 15, pp. 11137–11161, 2020.

[21] A. M. Shaker, M. Tantawi, H. A. Shedeed, and M. F. Tolba, "Generalization of convolutional neural networks for ECG classification using generative adversarial networks," *IEEE Access*, vol. 8, no. 1, pp. 35592–35605, 2020.

[22] S. H. Jambukia, V. K. Dabhi, and H. B. Prajapati, "Classification of ECG signals using machine learning techniques: a survey," in *Proceedings of the International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, March 2015.

[23] P. de Chazal and R. B. Reilly, "A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535–2543, 2006.

[24] T. Ince, S. Kiranyaz, and M. Gabbouj, "A generic and robust system for automated patient-specific classification of ECG signals," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1415–1426, 2009.

[25] W. Jiang and S. G. Kong, "Block-based neural networks for personalized ECG signal classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1750–1761, 2007.

[26] K. Luo, J. Li, Z. Wang et al., "Patient-specific deep architectural model for ECG classification," *Journal of Healthcare Engineering*, vol. 2017, no. 1, 13 pages, Article ID 4108720, 2017.

[27] A. S. Alvarado, C. Lakshminarayan, and J. C. Principe, "Time-based compression and classification of heartbeats," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1641–1648, 2012.

[28] C. Ye, B. V. Kumar, and M. T. Coimbra, "Heartbeat classification using morphological and dynamic features of ECG signals," *IEEE Transactions on Bio-Medical Engineering*, vol. 59, no. 10, pp. 2930–2941, 2012.

[29] Z. Zhang, J. Dong, X. Luo, K.-S. Choi, and X. Wu, "Heartbeat classification using disease-specific feature selection," *Computers in Biology and Medicine*, vol. 46, no. 1, pp. 79–89, 2014.

[30] L. Niu, C. Chen, H. Liu et al., "A deep-learning approach to ECG classification based on adversarial domain adaptation," *Healthcare*, vol. 8, no. 4, 2020.

[31] M. Alfaras, M. Soriano, and S. Ortín, "A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection," *Frontiers in Physics*, vol. 7, 2019.

[32] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[33] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[34] P. d. Chazal, "Detection of supraventricular and ventricular ectopic beats using a single lead ECG," pp. 45–48.

[35] I. A. American National Standards Institute, *Testing and Reporting Performance Results of Cardiac Rhythm and St Segment Measurement Algorithms*, ANSI, New York, NY, USA, ANSI/AAMI/ISO EC57, 2008.

[36] A. M. Alqudah, A. Albadarneh, I. Abu-Qasmieh, and H. Alquran, "Developing of robust and high accurate ECG beat classification by combining Gaussian mixtures and wavelets features," *Australasian Physical & Engineering Sciences in Medicine*, vol. 42, no. 1, pp. 149–157, 2019.

[37] H. Alquran, A. M. Alqudah, A. Al-Badarneh, and S. Almashaqbeh, "ECG classification using higher order spectral estimation and deep learning techniques," *Neural Network World*, vol. 29, no. 4, pp. 207–219, 2019.

[38] J.-H. Kim, S.-Y. Seo, C.-G. Song, and K. S. Kim, "Assessment of electrocardiogram rhythms by GoogLeNet deep neural network architecture," *Journal of Healthcare Engineering*, vol. 2019, Article ID 2826901, 10 pages, 2019.

[39] P. Molchanov, S. Tyree, T. Karras et al., "Pruning convolutional neural networks for resource efficient inference," 2016, http://arxiv.org/abs/1611.06440.