Efficient Learning of Healthcare Data from IoT Devices by Edge Convolution Neural Networks †

Yan He 1,2, Bin Fu 1,2,* , Jian Yu 1,2, Renfa Li 1,2 and Rucheng Jiang 1,2

1 College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; heyan1997@hnu.edu.cn (Y.H.); yujian@hnu.edu.cn (J.Y.); lirenfa@hnu.edu.cn (R.L.); jiangrc@hnu.edu.cn (R.J.)
2 Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha 410082, China
* Correspondence: fubin@hnu.edu.cn
† This paper is an extended version of our paper published in 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018.

Received: 25 November 2020; Accepted: 9 December 2020; Published: 15 December 2020

Abstract: Wireless and mobile health applications promote the development of smart healthcare. Effective diagnosis and feedbacks of remote health data pose significant challenges due to streaming data, high noise, network latency and user privacy. Therefore, we explore efficient edge and cloud design to maintain electrocardiogram classification performance while reducing the communication cost. These contributions include: (1) We introduce a hybrid smart medical architecture named edge convolutional neural networks (EdgeCNN) that balances the capability of edge and cloud computing to address the issue for agile learning of healthcare data from IoT devices. (2) We present an effective deep learning model for electrocardiogram (ECG) inference, which can be deployed to run on edge smart devices for low-latency diagnosis. (3) We design a data enhancement method for ECG based on deep convolutional generative adversarial network to expand ECG data volume. (4) We carried out experiments on two representative datasets to evaluate the effectiveness of the deep learning model of ECG classification based on EdgeCNN. EdgeCNN shows superior to traditional cloud medical systems in terms of network Input/Output (I/O) pressure, architecture cost and system high availability. The deep learning model not only ensures high diagnostic accuracy, but also has advantages in aspect of inference time, storage, running memory and power consumption.

Keywords: deep convolutional generative adversarial network; deep learning; edge computing; electrocardiograms

1. Introduction

With the economic prosperity and technological progress, wireless and mobile healthcare applications are proliferating [1]. Internet of things devices are widely used in human health monitoring. Due to the characteristics of streaming data, heterogeneous and high noise [2], the efficient analysis and diagnosis of health data face many challenges. The maturity of cloud computing and deep learning technology provides strong support for solving this problem. Nevertheless, there are still some problems in the application of deep learning model based on cloud platform for real time health monitoring. First, continuous transmission of health data will consume a lot of network resources. Second, the return time of cloud decision is uncertain due to the fluctuation of network transmission. Last, users’ health data are stored in the cloud, which will involve personal privacy issues. Therefore, data-intensive analysis in smart healthcare requires a new computational model to provide location-aware and delay sensitive monitoring with intelligent and reliable control [3].

Electrocardiogram (ECG) is one of the most commonly used examinations in clinic. Remote ECG monitoring plays an important role in early diagnosis and prevention of cardiovascular diseases.
This usually requires low-cost, high-convenience and low-latency services for ECG users. With the application of deep learning in the field of ECG, the automatic detection of ECG has made great progress [4]. At present, most of the training data used by researchers are MIT-BIH [5] and other public datasets, which have certain limitations in data volume and category. For example, the MIT-BIH dataset is sufficient in a few kinds of heart disease data, while the category of disease data is not enough to meet the needs of deep learning model training. As a result, research on ECG data is far from reaching the practical application.

In this work, we take advantage of both edge and cloud to design an efficient health monitoring architecture. We use convolutional neural networks (CNN) to develop a streamlined and efficient model to identify ECGs on the edge. We noticed the imbalance of ECG data and its influence on diagnostic accuracy. Specifically, the contributions of this work can be summarized as follows.

Firstly, a new hybrid intelligent healthcare architecture based on edge computing and cloud computing, named EdgeCNN, is proposed. This architecture can flexibly learn medical data from edge devices. Deep learning model is deployed to run on edge devices, which makes analysis and diagnosis closer to Input/Output(IoT) data sources. This can significantly relieve learning latency and network I/O and reduce the pressure of large user groups and massive data on the cloud platform.

Secondly, relying on the EdgeCNN architecture, we design an efficient ECG edge computing diagnosis model and learning algorithm. Then, we successfully deploy it on edge devices. The model can infer ECG in real time closer to data source and get a good balance between diagnosis accuracy and resource loss. Experimental results show that, compared with pure cloud computing architecture, EdgeCNN can not only ensure reasonable accuracy, but also have obvious advantages in diagnosis delay, network I/O, application availability and resource cost. More importantly, it can effectively protect the privacy of user data from IoT devices.

Thirdly, we propose a method to enhance ECG by deep convolutional generative adversarial network (DCGAN) [6]. DCGAN is a stable network architecture based on CNN extension, which has a very high credibility in unsupervised learning. Data enhancement can provide enough data quantity and category for ECG diagnosis. The experimental results show that, after the data enhancement of ECG data related categories, not only the overall accuracy of the deep learning model is improved, but also the ECG categories that the model can diagnose are greatly expanded, which greatly improves the practical usability of the system.

This paper is an extended and enhanced version of an earlier conference paper published in IEEE 24th International Conference on Parallel and Distributed Systems [7]. Our initial conference paper does not address the problem of the imbalance of ECG data and its impact on diagnostic accuracy. This manuscript addresses this issue and provides a data enhancement method for ECG based on DCGAN. Experiments are presented to verify the effectiveness of the data enhancement.

The rest of the article is organized as follows. Section 2 reviews related works. Section 3 introduces the hybrid intelligent healthcare architecture and the ECG diagnosis model. Section 4 presents the method to enhance ECG data by DCGAN. Section 5 analyzes the experimental results. We draw conclusions and note future work in Section 6.

2. Related Work

Health informatics has recently become an important area of concern and one of the great challenges in engineering. Mobile Edge cloud is an emerging field, which is rapidly applying across the border. Cloud computing can help the IoT to calculate and analyze data. Edge devices are important medical tools for timely monitoring and preventing cardiovascular diseases. Obviously, the tendencies of combining health informatics with mobile edge cloud computing and developing ECG devices based on IoT are inevitable. Antonio et al. [8] reviewed the evolution of the role that electrophysiology plays as part of occupational health. They summarized the benefits in the development of wearable and smart devices for cardiovascular monitoring and their possible applications, demonstrating the trends of using mobile ECG devices across different environmental settings and populations.
Khairuddin et al. [9] not only reviewed previous works on conventional ECG devices to identify their limitations, but also provides some insights into how the IoT has the potential to develop medical applications in ECG devices. Liu et al. [10] combined mobile cloud computing and health informatics field to develop a system for ECG extraction and analysis using mobile computing, which they plan to deploy the system in commercial environment.

Using deep learning to classify and analyze medical images is popular. Pereira et al. [11] used CNN to recognize handwritten images to help identify the early stages of Parkinson’s disease. Their model learns features from the signals of the smart pen, which uses sensors to capture the movements of handwriting during personal exams. Lipton et al. [12] studied the performance of LSTM in analyzing and identifying multivariate time series patterns of medical measurements in the intensive care unit. Daniele et al. [13] wrote a survey to review the deep learning in health informatics.

At the same time, many researchers use deep learning to diagnose ECG. Hannun et al. [14] collected ECG data from 53,549 patients and designed a 34-layer CNN model to classify the 12 heart rhythm irregularities. The final classification accuracy is higher than the diagnosis accuracy of human experts. Chauhan et al. [15] used LSTM in a recurrent neural network to diagnose ECG. This method allows ECG signals to be directly input into the network without any complicated data preprocessing. Adam et al. [16] used a neural network to extract a complex segment of QRS waves of ECG signals, and then used it for user authentication. The user authentication accuracy of this system is as high as 99%. In the medical field, the ECG classification challenges are held every year. Zihlmann et al. [17] constructed two models, CNN and LSTM, and used the PhysioNet/CinC 2017 dataset for training and testing. In the comparison of results, the LSTM model achieved a $F_1$-score [18] of 82.1% accuracy, which is better than the CNN model. Wang et al. [19] designed a five-class model DMSFNet, which uses 12-lead CPSC 2018 and PhysioNet/CinC 2017 dataset to classify atrial fibrillation. Zhang et al. [20] designed STA-CRNN model with attention mechanism for eight classification and obtained an $F_1$-score of 0.835. The training and reasoning of these ECG diagnosis models are completed in the resource rich server or cloud.

Because heart disease has the characteristic of occurring suddenly, it is necessary to use deep learning in mobile devices for real-time monitoring. There are many studies which use mobile applications to perform task, such as using CNN to identify garbage in images [21]. However, experiments show that the resources consumption of these applications are still very high. It takes 5.6 s to return the prediction results, but it consumes 83% of CPU and 67 MB of memory. Amato et al. [22] ran CNN on the RaspberryPi board and integrated a smart camera to find empty parking spaces. Ravi et al. [23] demonstrated the development of a mobile device fitness application that uses deep learning to classify human activities. However, DNN models in resource-constrained devices often have fewer hidden layers, so their performance is poor. Nguyen et al. [24] proposed a conceptual hardware and software framework to support intelligent IoT applications.

Zhang et al. [25] enhanced the EEG data with DCGAN and compared it with traditional methods including geometric transformation (GT), autoencoder (AE) and variational autoencoder (VAE). The results show that DCGAN obtains better enhancement performance. Zanini et al. [26] used DCGAN to enhance EMG to study Parkinson’s disease. This model expands the patient’s tremor dataset by learning the patient’s different tremor frequency and amplitude, and extends it to different exercise program sets. Cao et al. [27] proposed a novel data augmentation strategy based on duplication, concatenation and resampling of ECG episodes to balance the number of samples among different categories as well as to increase the diversity of samples. Salamon et al. [28] used a variety of audio data enhancement techniques and discussed the impact of different enhancement methods on the performance of the proposed CNN architecture. Perez explored and compared various solutions to the problem of data augmentation in image classification [29], which attempted to use Generative Adversarial Network (GAN) to generate images of different styles, and discussed the success and inadequacies of the method on various datasets.
3. ECG Diagnosis Model Based on Edge Cloud Intelligent Medical Architecture

3.1. The Edge Cloud Intelligent Medical Architecture

The intelligent medical system based on cloud cannot meet the urgent needs of people for daily health monitoring in the era of IoT. In this section, we introduce a hybrid intelligent medical architecture called EdgeCNN (see Figure 1) to solve existing problems. EdgeCNN realizes intelligent collaboration of edge and cloud. The architecture composes of IoT device layer, edge computing layer, cloud platform layer and third-party service layer.

The IoT layer, which involves a series of health devices such as electrocardiographs, smart wristbands, smart watches, blood pressure meters and so on, is used to monitor various physiological indicators of human body and produce health data. Due to hardware limitations in computing power, storage and power, IoT devices do not have the ability to efficiently process these data [30]. Users need to upload data to the cloud or carry data to the hospital to obtain health analysis results. However, both methods are not real-time and convenient, thus cannot give early warning for potential health problems.

The edge layer run the computing tasks on the resources close to the data source, so as to effectively reduce the delay of the computing system, reduce the data transmission bandwidth, and protect the data security and privacy. The edge layer is responsible for the execution of edge AI reasoning, executing the AI model distributed from the cloud, and feeding back the execution results to the user or cloud. In our architecture, the IoT monitoring device transmits the monitored data to an edge device through a local area network (e.g., WLAN, BlueTooth, and ZigBee), where a deep learning model for processing the corresponding data is deployed for diagnosis. Note that the deep learning model for edge computing needs to be re-designed, to make a trade-off between the accuracy of the diagnosis and the complexity of the model [31] so that it can run efficiently at the edge devices. By introducing the edge computing node, the data are transmitted only in the user-controllable local area network, thus protecting user’s data privacy.

The cloud platform layer provides global, non real-time, long-term big data processing and analysis. It carries out centralized AI model training according to business requirements, historical data, real-time data and AI execution feedback and sends the AI model to edge nodes for execution. In the framework of cloud and edge integration, it is very important to study the interaction between clouds and edge. Although users can obtain efficient health monitoring and diagnosis using only IoT devices and edge computing devices, they can still upload a part of data to the cloud platform for
more accurate inference. As mentioned above, the deep learning model deployed on the edge device is a trade off between accuracy and complexity.

The top layer of the architecture is the third-party service. Users can authorize relevant hospitals or private doctors to obtain their own data through cloud platform for further detailed diagnosis. The cloud platform can intelligently connect patients, medical staff, medical service providers and insurance companies in a seamless and collaborative way, so that patients can experience one-stop nursing, medical and insurance services.

3.2. Diagnosis Model for Agile Learning in EdgeCNN

Although some of the research based on deep learning algorithms have been able to perform electrocardiogram diagnostics well, their models are too complex to be deployed on edge devices for low latency diagnosis. Diagnosis cannot effectively promote the popularization of home-based smart medical. In this section, we introduce a model that can effectively trade off between accuracy and complexity so that it can be deployed on smart devices to perform low latency ECG diagnosis for users.

The general structure of our model is shown in Figure 2. This is a convolutional neural network with five convolutional layers and one fully connected layer. Since the length of the ECG signal in the dataset is different, we preprocess the original data, splitting the ECG data and inputting the model with one-dimensional data. The data are batched into the CNN network, a convolution operation is performed on the input data using a $1 \times 5$ convolution kernel and the offset value is added after this.

![Figure 2. The proposed neural network architecture.](image-url)
To enhance the generalization of the network and enable rapid convergence of model training, we use batch-normalization (BN) [32] to normalize the data after each convolution. The BN can normalize input sample features to make the data be distributed with a mean of 0 and a standard deviation of 1. If we do not normalize the data, due to the scattered distribution of the sample features, the neural network will learn slowly or even have difficulty learning. The formula for data normalization is given in Equation (1):

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}},$$

where $x^{(k)}$ represents the $k$th dimension of the input data, $E[x^{(k)}]$ represents the average of the dimensions and $\sqrt{Var[x^{(k)}]}$ represents the standard deviation. However, if it is done simply, it will reduce the layer of expression. Thus, BN adds two parameters ($\gamma$ and $\beta$) to maintain the expressiveness of the model. The form is Equation (2):

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}.$$

Rectified linear activation (ReLU) [33] can save computational complexity, alleviate overfitting problems to a certain degree and obtain faster convergence than other activation functions. For ReLU functions, the formula is Equation (3):

$$ReLU(x) = \max(0, x).$$

Its function is to make the calculated value equal to 0 if it is less than 0, otherwise it remains the same. It has been proven in practice that the trained network is fully moderately sparse. The visualized training results are similar to the pre-trained results of the traditional methods. This also shows that ReLU has the ability to guide moderate sparseness.

The pooling layer can reduce the parameters and calculations while retaining the main features, prevent over-fitting and improve the model’s generalization ability. The model uses $1 \times 2$ average pooling, which averages the two values in the domain. This can reduce the error of the increase in the variance of the estimate caused by the limited size of the neighborhood. The fully connected layer converts the output from the convolutional layer into a $n \times T$ vector. Each number corresponds to a category (N, A, O or ∼). Finally, softmax is used to obtain the most likely prediction result. The performance of softmax is Equation (4):

$$S_i = \frac{e^{V_i}}{\sum_j e^{V_j}},$$

where $V_i$ represents the $i$-th element in the array $V$. When a sample passes through the softmax layer and outputs a vector of $1 \times T$, the index of the one with the highest value in the vector is taken as the prediction label of the sample.

However, actual ECG data are generated continuously over time and are not similar to fixed-length ECG data when training models. The actual process of real-time diagnosis is abstracted as Algorithm 1. We note that the time series data generated during a certain period of time are set $X = \{x_1, x_2, \ldots, x_n\}$. The number of data points that the monitoring device can generate per second is $m$, and the time length of the time series data $X$ is $n/m$ seconds. We use $X$ as the input of the model, and then the required data length for one prediction is $n$ and the total time for the model to predict $X$ is $t$. To prevent data congestion, it is necessary to satisfy $t \leq (n/m)$. 
Algorithm 1 Real-time diagnosis algorithm

Require: Time series data: \( \{x_1, x_2, \cdots, x_i\} \)

Ensure: Real-time diagnosis, no data congestion.

while True do
  if \( i < n \) then
    Save previous data and continue to receive data
  else
    \( X = \{x_1, x_2, \cdots, x_n\} \)
    Call model to diagnose \( X \) which takes time \( t \)
    if \( (n/m) < t \) then
      Cause data congestion
      Break
    else
      Output diagnosis result
    end if
  end if
end while

4. Data Augmentation

ECG automatic diagnosis using deep learning needs a lot of training data. At present, the lack and imbalance of ECG data have become an obstacle to the development of ECG automatic diagnosis technology. In this section, we design a data enhancement method for ECG based on DCGAN to expand ECG data volume and data categories. In our research field, we are the first to use DCGAN to try to solve the ECG imbalance problem.

4.1. Analysis of MIT-BIH Datasets

MIT-BIH arrhythmia database is an authoritative public dataset in ECG research field, which has rich data categories and accurate annotation specification. However, MIT-BIH data distribution is very unbalanced; only a few categories of data can meet the requirements of deep learning training. There are 23 data labeling categories in the MIT-BIH arrhythmia database, but only seven categories have thousands of samples. This has led to the current ECG automatic diagnosis methods being unable to expand the scope of diagnosis, hindering its pace of practical application. Table 1 shows the top ten label categories for the number of beats.

| Comment Code | Description                                      | Number of Heartbeats |
|--------------|--------------------------------------------------|----------------------|
| 1            | Normal beat                                      | 75,019               |
| 2            | Left bundle branch block beat                    | 8072                 |
| 3            | Right bundle branch block beat                   | 7255                 |
| 5            | Premature ventricular contraction                | 7129                 |
| 12           | Fusion of paced and normal beat                  | 7024                 |
| 8            | Atrial premature beat                            | 2546                 |
| 28           | Rhythm change                                    | 1250                 |
| 38           | Fusion of paced and normal beat                  | 982                  |
| 6            | Fusion of ventricular and normal beat            | 802                  |
| 31           | Ventricular flutter beat                          | 472                  |

It can be seen from the distribution of the table data that the total number of heartbeat data in the top five categories exceeds 7000, which meets the basic requirements of deep learning model training. In actual research, scholars have mostly studied these five categories. However, the other categories of ECG diagnostic classification are rarely considered in current research. To overcome the current
bottleneck of insufficient ECG diagnosis data, we try to augment the heartbeat data labeled 8, 38, 6 and 31 through DCGAN, and observe the data generation process. The original heartbeat samples of these four types of ECG after denoising and QRS wave extraction are shown in Figure 3.

Figure 3. Sample of heartbeat data labeled 8, 38, 6 and 31: (a) Atrial premature, (b) Fusion of paced and normal beat, (c) Fusion of ventricular and normal beat, (d) Ventricular flutter beat.

4.2. Model Building

The framework of GAN includes a generative model $G$ and a discriminative model $D$, which makes $G$ learn the distribution of the data [34]. GAN is a zero-sum game mechanism, generating networks and discriminating networks against each other. The generating network tries to generate realistic fake samples to deceive the discriminating network as much as possible, and the discriminating network needs to identify whether a sample is fake or real as often as possible. The process can be seen in Figure 4.

Figure 4. The structure of generating ECG in GAN.
Random noise usually obeys the Gaussian distribution. The generation network generates fake samples on the basis of random noise. When the discriminator cannot distinguish between the real image and the generated image, the network is fit. The loss function of the two networks can be described as:

$$\min_G \max_D V(G, D) = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log(D(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] .$$ (5)

For the generator \( G \), to generate fake samples that can deceive the discriminator \( D \), it is necessary to minimize the value of \( \log(1 - D(G(z))) \), that is, to maximize the discriminant probability \( D(G(z)) \) of the generated sample. In actual training, the method is alternative training of generators and discriminators.

Although GAN has acquired huge success, it has been proved to be unstable to train and produce nonsensical outputs in the generator. Thus, DCGAN was proposed in 2015. CNN performs well in supervised learning and worse in unsupervised learning, while DCGAN performs well in unsupervised learning. We can combine CNN in supervised learning with GAN in unsupervised learning to get better performance compared with GAN. DCGAN makes the following adjustments. First, replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator). Second, use batchnorm in both the generator and the discriminator. Third, remove fully connected hidden layers for deeper architectures. Fourth, use ReLU activation in generator for all layers except for the output, which uses Tanh. Fifth, use LeakyReLU activation in the discriminator for all layers. The structure of DCGAN is shown in Figure 5.

4.3. Data Generation

In this section, four types of ECG data and randomly generated noise sequences are used as input of DCGAN, which is used for training and data generation. The input of the generator consists of a series of data sequences, each of which consists of 250 noise points. The output is the generated ECG sequence, whose length is set to 250. The input of the discriminator is the result produced by the generator and the real ECG data.

The learning rate of the Adam optimizer in the experiment is 0.002. Since the BN mechanism is used, the batch size for each batch training is set to 64. According to the data characteristics of ECG, one-dimensional convolution and deconvolution methods are used in the process of training using tensorflow. Each iteration operation sets an event tracking to record its discriminative network loss value and generative network loss value. After 50 epochs of training, the generating samples of the four types of ECG beats are shown in Figure 6. They are the examples of atrial premature beats, ventricular fusion heartbeat, pacing mixed heart rate, and ventricular flutter. It can be seen in the figure that, although the generated pattern is not as smooth as the original data, the important basic features of the original data have been reflected.

In the whole training process, the change trend of the loss value \( d_{\text{loss}} \) of discriminative network and the loss value \( g_{\text{loss}} \) of the generative network is shown in Figure 7. It can be seen that, in the whole process, the discriminative network and the generative network are undergoing alternating and iterative optimization, which is in line with the idea of zero-sum game. The optimization goal
of GAN is mainly composed of two parts: the probability of judging whether the real sample is true by the discriminative network and the probability of judging whether the generated sample is true by the generative network. This makes them have a confrontation game in the process of training. The generative network tries to generate more realistic data to deceive the discriminative network and the discriminative network tries to distinguish the real samples from the generated samples. Therefore, in terms of optimization goals, the discriminative network wants $V(G, D)$ to be as large as possible, while the generating network wants $V(G, D)$ to be as small as possible. This leads to the phenomenon of iterative optimization in the training process.

Figure 6. Sample ECG beat generation: (a) Atrial premature, (b) Fusion of ventricular and normal beat, (c) Fusion of paced and normal beat, (d) Ventricular flutter beat.
Figure 7. Alternating and iterative optimization of discriminant and generative networks: (a) g_loss, (b) d_loss.

5. Evaluation

In this section, we use the deep learning model in EdgeCNN to perform feature learning and classification on the data in the PhysioNet/CinC 2017 and MIT-BIH datasets. The cardiology challenge announced by PhysioNet in 2017 is to classify atrial fibrillation (AF) with short single lead electrocardiograms. The incidence of AF is 1–2%, increasing in the incidence of the growing age, is the most common persistent arrhythmia, which have considerable morbidity and mortality. Our experiments utilize their published dataset including 8528 single-lead ECG recordings, 9–60 s in length, with a sampling rate of 300 Hz, and it is divided into four categories: normal rhythm (N), AF rhythm (A), other rhythm (O) and noisy recordings (∼). Examples of these different types of electrocardiograms are shown in Figure 8. During the experiment, due to the problem of MIT-BIH data imbalance, only a small part of the categories can be fully trained. To solve this problem, we enhance the MIT-BIH dataset.

Figure 8. Typical recordings for each of the 4 classes in the dataset: (a) Normal rhythm (N), (b) AF rhythm (AF), (c) Other rhythm (O), (d) Noisy recordings (∼).
5.1. Evaluation of EdgeCNN

At first, we evaluate the performance of EdgeCNN diagnosis model in PhysioNet2017 dataset based on the accuracy of prediction, the cost of storage and memory and prediction time. Afterwards, the comparative experiments on two diverse architectures for ECG diagnosis are run, one that utilizes the hybrid architecture in EdgeCNN and another that only utilizes cloud platform.

5.1.1. Performance of the ECG Diagnosis Model Deployed on the Edge Devices

We evaluate the overall performance of the ECG diagnosis model that is deploy on the edge devices of processor snapdragon 660. After testing the prediction accuracy of normal rhythm (N), AF rhythm (A), other rhythm (O) and noisy recordings (~), the result is shown in Figure 9, which shows the prediction accuracy rate of “Normal” is 87%, “AF” is 84%, “Other” is 81% and “Noisy” is 60%.

The figure describes that there are three different categories of ECG which are easily misjudged as “Other”. In addition, the ratio of “AF” to “Other” is the same, 11%. “Noisy” has the lowest prediction accuracy, because the majority of them are classified as “Other” and “Normal”. In the true production environment, to avoid the transmission of numerous amount of data in the network environment, most of the predictions of the hybrid architecture are completed at the edge devices. This strategy can decline the ratio of noise generated by data transmission and decrease the predicted error to some degree.

Then, we compare the accuracy of the model in predicting with $F_1$-score, the cost of storage and memory and predict time. $F_1$-score is a measure of the accuracy of the model in predicting atrial fibrillation. Its calculation formula is Equation (6):

$$F_1 = \frac{1}{4} \times \left( \frac{2 \times N_n}{\Sigma N + \Sigma n} + \frac{2 \times A_a}{\Sigma A + \Sigma a} + \frac{2 \times O_o}{\Sigma O + \Sigma o} + \frac{2 \times P_p}{\Sigma P + \Sigma p} \right).$$

(6)

The definition of each variable is shown in Figure 10.

![Figure 9. The prediction accuracy for each category.](image)

![Figure 10. Counting rules for the numbers of the variables.](image)
The results of the ECG diagnosis model and the optimal model in 2017CinC [35] is shown in Table 2. Bo et al. [35] used a CNN with a 16-layer convolutional layer and obtained a final score of 0.83. However, this complex model structure needs a larger volume and causes higher computational complexity. Although our streamlined and efficient model acquires a score of 0.78, which is a little bit lower than the optimal model, the optimal model’s physical storage space reached 4.95 MB after being persisted, which is almost 1.5-fold larger than our model. Moreover, the memory at runtime reached 74 MB, which is 23 MB more than our model.

| Attributes         | EdgeCNN Model | 2017CinC Model |
|-------------------|---------------|---------------|
| $F_1$             | 0.78          | 0.83          |
| Model size        | 3.30 MB       | 4.95 MB       |
| Memory            | 51 MB         | 74 MB         |
| Predict time      | 310–330 ms    | 430–490 ms    |

The definition of prediction time is the time which is required for the ECG diagnosis model from receiving data to output the final prediction result. It means that the faster diagnosis speed can decrease the general delay of the system. Hence, we should ensure a shorter prediction time and acquire the basic accuracy which is higher than cardiologists. The complex calculations make the prediction time of 2017CinC optimal model about 100 ms longer than our model. Such a long delay is definitely catastrophic in the field of smart healthcare.

Therefore, our model absolutely occupies an advantage from the perspective of the size of the storage space occupied by the model, the memory occupied by the runtime and the inferred time, which makes the model easily deploy on resource-constrained edge devices. It can fit the new architecture of the cloud-edge-integrated home smart healthcare proposed by us.

5.1.2. Comparison of Delay on Two Different Architectures

We build two system with diverse architectures for ECG diagnosis, one system uses a hybrid architecture in EdgeCNN while the other uses cloud platform. The experiments compare the two architectures in terms of delay. Delay is an important indicator that is used to evaluate the effectiveness of a system. We measure the length of time from data generation to the feedback of diagnostic results to the user under different request volumes. It contains both data transfer time and model prediction time.

The predicted total time $t$ is mainly composed of four parts: the data transmission delay $t_1$, the data propagation delay $t_2$, the time $t_3$ from when the model receives the data to the final output of the prediction result and the time $t_4$ when the prediction result is returned to the user. Therefore, time $t$ can be described as follows:

$$t = t_1 + t_2 + t_3 + t_4.$$  (7)

Since the propagation delay $t_2$ is related and proportional to the network distance between the client and the server and $t_4$ is very small, according to Algorithm 1, to avoid data congestion, $t$ needs to meet the conditions $t \leqslant (n/m)$, which must be considered from two aspects that affect $t_1$ and $t_3$:

1. When the number of users gradually increases, the amount of data generated in the same time period will also increase. Whether the network bandwidth of the data processing point is sufficient to transmit such a large amount of data within a limited time will directly affect the sending delay $t_1$.
2. The performance of the data processing point is limited. When the data that need to be processed at the same time reach the performance bottleneck of the data processing point, the next part of the data will be queued for processing, which will increase the time $t_2$. 
Assuming that the bandwidth of the processing point is 1 Mbps, and $N$ is used to represent the number of data segments generated in the same time period, the transmission time $t_1$ of these data transmission in the network can be calculated by Equation (8):

$$t_1 = \frac{N \times (n \times 4) / 1024}{(a \times 1024) / 8}.$$  \hspace{1cm} (8)

If the hardware performance is sufficient to handle the current concurrency, $t_2$ will be constant, and then obviously the time $t$ will be inversely proportional to the bandwidth $a$. The larger is the bandwidth, the smaller is the sending delay $t_1$. The overall performance of data processing points will vary with changes in hardware, and quantitative measurements cannot be made. However, as the amount of data that need to be processed at the same time increases, the average number of requests processed by each thread in the processing point will inevitably increase. We can measure its impact on time $t$ under the condition that the requests processed by a single thread gradually increase.

As shown by the experimental results in Figure 11, as the request volume of each thread gradually increases, the cloud faces bottlenecks in performance and network I/O. As a result, there is a gradual growth in latency. The delay is about 393 ms when the number of concurrent requests to be processed by each thread of the cloud platform is less than 7. The delay reaches at least 384.7 ms when the number of simultaneous requests processed by the thread is 7 or 8. Because of the request rate and the fastest rate handled by a single thread of the cloud platform, which are close at the moment, the thread does not need to be awakened frequently, and the cloud platform has the highest processing efficiency. Once the requests exceed the maximum number of requests that a single thread can afford, they have to be queued. The consequences are that it will result in a large amount of data congestion and the delay will rise until the real-time diagnostic capability lose.

In contrast, since the edge devices are distributed and deployed in each user’s home, it can be the user’s smart mobile phone, a gateway or router, a smart device customized for smart healthcare or other intelligent electronic devices. All of them contain certain data computing capabilities.

With the gradual growth of the amount of users, the problems which are similar to the data congestion in the cloud platform will not happen, and the delay will continuously keep stable at about 320 ms. There is usually at least one user at home, which presents a stable and efficient solution to handle their healthcare data.

![Figure 11. Comparison of the delays of the two system architectures.](image)

5.1.3. Network I/O and High Availability

Due to the low dependence on the cloud platform, the data generated by IoT does not need to be transmitted to the cloud platform in real time through the network. A portion of the data is only uploaded when the user authorizes the use of a third-party service. This makes the cloud platform not encounter too much dilemma on the network I/O.
Moreover, for the traditional cloud-only architecture, it is difficult to ensure that the cloud platform does not have problems. Once the cloud platform fails, all users’ services will be interrupted, which is catastrophic for the user experience and the system platform. The hybrid architecture with cloud and edge is a good way to circumvent this problem. Each edge node has autonomy. Even if the cloud platform fails, most users can still rely on the edge device to obtain services normally, and only special third-party services will be suspended. This ensures high availability of the entire architecture.

The performance comparison of the two architectures, EdgeCNN architecture and traditional cloud based platform architecture, on different metrics are summarized in Table 3. First, because the processing unit is closer to the data source, the hybrid architecture achieves lower and more stable latency without data congestion due to the large user volume. Second, one of the important reasons for data congestion is the bottleneck of the network I/O. When large amounts of data flock to the cloud platform in real time, network I/O is under tremendous pressure, resulting in a decline in real-time processing power of the system. The hybrid architecture circumvents this problem by spreading the data across the various edge compute nodes. Third, the hybrid architecture uses edge computing nodes to diagnose, which greatly reduces the cost of operation and maintenance of cloud platforms. Fourth, the hybrid architecture can ensure that data are transmitted within the user-controllable local network, avoiding all data being exposed to the Internet, effectively protecting user privacy. Fifth, the hybrid architecture takes distributed edge nodes to process data. It will not paralyze the whole system due to the failure of the cloud platform. The high availability of the system is guaranteed.

In addition to the related issues mentioned above, power consumption is also a key issue that edge devices need to consider. To this end, we measured the power consumption of the two models when running on the experimental equipment, as shown in Figure 12. We can see from the figure that it takes 325 min for the remaining power to drop from 100% to 70% without any processing tasks. After deploying the 2017CinC model, the time required for the remaining power to drop to 70% was suddenly shortened by 75 min. It can be seen that this reduces the overall endurance of the device by approximately 250 min. EdgeCNN shortens the time required for the remaining battery to drop to 70% by 25 min. Compared with the former, the battery life has increased by 50 min, and the overall battery life can be increased by about 167 min. It can be seen that EdgeCNN has obvious advantages in energy efficiency.

### Table 3. Comprehensive comparison of the two architectures.

| Attributes       | Our Architecture | Traditional Cloud Architecture |
|------------------|-------------------|-------------------------------|
| Delay            | Low, Stable       | High, Unstable                |
| Network I/O      | Extremely few     | High pressure                 |
| Cost             | Less              | High                          |
| Privacy          | More secure       | Insecure                      |
| High Availability| Satisfy           | Not satisfied                 |

Figure 12. Power consumption comparison.
5.2. Evaluation of Data Augmentation

In this section, we evaluate the performance of our proposed ECG diagnosis model in MIT-BIH dataset. Figure 13 shows the main process of ECG data identification. We compare the diagnostic accuracy of the model before and after the data enhancement. The confusion matrix of the test results before data enhancement is shown in Figure 14. It can be seen from the table that the four categories of normal beat (1), left bundle branch block (2), right bundle branch block (3) and ventricular premature beats (5) with sufficient data volume have high diagnostic accuracy. The four categories of atrial premature beats (8), pacing mixed heart rate (38), ventricular fusion heartbeat (6) and ventricular flutter (31) decrease with the amount of data, and the diagnostic accuracy also drops significantly, until the diagnosis goes down to zero. The confusion matrix of the test results after data enhancement is shown in Figure 14. It can be seen from the matrix that almost all ECG categories are most likely to be misclassified as ventricular premature beats, which indicates that there may be some common features of ventricular premature beats. Comparison with Figure 14 shows that data enhancement plays an important role in improving the overall accuracy.

![Figure 13. The main process of ECG data identification.](image)

![Figure 14. Confusion matrix: (a) Confusion matrix of test results before data enhancement, (b) Confusion matrix of test results after data enhancement.](image)

Figure 15 shows the change in diagnostic accuracy of eight ECG categories before and after data augmentation. It can be seen that, before the data augmentation, although the diagnostic accuracy of the four ECGs of normal beat, left bundle branch block, right bundle branch block and ventricular premature beats have a good performance of more than 90%, the diagnostic accuracy of the three types of pacing mixed heart rate, ventricular fusion heartbeat and ventricular flutter are very poor. The diagnostic accuracy of pacing mixed heart rate is as low as 23.14%, while ventricular
fusion heartbeat and ventricular flutter cannot even be classified at all. After the data are enhanced, the diagnostic accuracy of the unenhanced data categories is more than 93%, which have a slight fluctuations compared with the previous ones. However, the fluctuations are within the normal range. The diagnostic accuracy of the enhanced data categories have undergone qualitative changes. Among them, the pacing mixed heart rate has increased from 23.14% to 84.46%, and the increase rate has reached 265%. Ventricular fusion heartbeat and ventricular flutter have, respectively, 79.17% and 95.97% diagnostic accuracy. Experimental data show that data enhancement has a substantial meaning for the classification of ECG diagnosis. The expansion of diagnosis categories and the improvement of accuracy make ECG diagnosis more practical.

Figure 15. Comparison of experimental results before and after data enhancement.

6. Conclusions

Smart medical treatment can provide timely and effective services for people’s health. In this article, we present a hybrid smart medical architecture based on edge and cloud computing. This architecture can not only reduce diagnosis delay and transmission overhead, but also protect data privacy. Furthermore, we design an effective deep learning model for ECG inference which can be deployed and run on edge smart devices. This diagnosis model achieves a good balance between diagnosis accuracy and resource cost. In addition, we propose a data enhancement method for ECG based on DCGAN, which expands ECG data volume and data categories. The experimental results show that, through data enhancement, not only the overall accuracy of the diagnosis model is improved, but also the diagnosis category is expanded.

Further work can be carried out from the following aspects. (1) The representation of ECG’s medical morphological features in data still needs to be further explored. Some heart diseases can be judged by a simple and single ECG waveform, but the waveforms of many heart diseases often show complex and diverse morphological characteristics. This requires researchers to have a deeper understanding of these data to design more excellent diagnostic model. (2) Although data enhancement technology has effectively improved the diagnostic accuracy of some ECG categories, most of the ECG category data are still vacant. This requires further strengthening of cross-professional cooperation, especially cooperation with hospitals with massive data sources. (3) The model needs to consider individual differences. The current ECG diagnosis is mostly based on publicly available datasets and does not fully consider individual differences. To further improve the practical application effect of ECG automatic diagnosis model, it is necessary to update the diagnosis model according to the characteristics of each individual.

Author Contributions: Conceptualization, J.Y. and Y.H.; formal analysis, B.F. and Y.H.; funding acquisition, Y.H. and B.F.; investigation, B.F. and Y.H.; methodology, Y.H. and J.Y.; Project administration, R.L.; resources, B.F. and Y.H.; software, Y.H.; validation, B.F. and Y.H.; writing—original draft, B.F. and Y.H.; and writing—review and editing, Y.H., R.L. and R.J. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the National Natural Science Foundation of China under Grant 61932010.
Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stankovic, J.A. Research Directions for Cyber Physical Systems in Wireless and Mobile Healthcare. *ACM Trans. Cyber Phys. Syst.* 2016, 1. [CrossRef]
2. Chen, M.; Mao, S.; Zhang, Y.; Leung, V.C.M.; Li, G. Big data related technologies, challenges and future prospects. *Inf. Technol. Tour.* 2014, 15, 283–285.
3. Tang, B.; Chen, Z.; Hefferman, G.; Pei, S.; Wei, T.; He, H.; Yang, Q. Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities. *IEEE Trans. Ind. Inform.* 2017, 13, 2140–2150. [CrossRef]
4. Murat, F.; Yıldırım, Z.; Talo, M.; Baloglu, U.B.; Acharya, U.R. Application of deep learning techniques for heartbeats detection using ECG signals- Analysis and Review. *Comput. Biol. Med.* 2020, 120, 103726. [CrossRef]
5. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* 2002, 20, 45–50. [CrossRef]
6. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* 2015, arxiv:1511.06434.
7. Yu, J.; Fu, B.; Cao, A.; He, Z.; Wu, D. EdgeCNN: A Hybrid Architecture for Agile Learning of Healthcare Data from IoT Devices. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018.
8. Baldassarre, A.; Mucci, N.; Padovan, M.; Pellitteri, A.; Viscera, S.; Lecca, L.I.; Galea, R.P.; Arcangeli, G. The Role of Electrocardiography in Occupational Medicine, from Einthoven’s Invention to the Digital Era of Wearable Devices. *Int. J. Environ. Res. Public Health* 2020, 17, 4975. [CrossRef]
9. Khairuddin, A.M.; Azir, K.N.F.K.; Kan, P.E. Limitations and future of electrocardiography devices: A review and the perspective from the Internet of Things. In *Proceedings of the 2017 International Conference on Research and Innovation in Information Systems (IÇRIIS)*, Langkawi, Malaysia, 16–17 July 2017.
10. Liu, Y.; Abbasi, A.A.; Aghaei, A.; Abbasi, A.; Al-Qaness, M.A.A. A Mobile Cloud-Based eHealth Scheme. *Comput. Mater. Contin.* 2020, 63, 31–39. [CrossRef]
11. Pereira, C.R.; Pereira, D.R.; Papa, J.P.; Rosa, G.H.; Yang, X.S. Convolutional Neural Networks Applied for Parkinson’s Disease Identification. In *Machine Learning for Health Informatics: State-of-the-Art and Future Challenges*; Holzinger, A., Ed.; Springer International Publishing: Cham, Switzerland, 2016; pp. 377–390. [CrossRef]
12. Lipton, Z.C.; Kale, D.C.; Elkan, C.; Wetzel, R. Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv* 2017, arxiv:1511.03677.
13. Ravi, D.; Wong, C.; Deligianni, F.; Berthelot, M.; Andreu-Perez, J.; Lo, B.; Yang, G.Z. Deep Learning for Health Informatics. *IEEE J. Biomed. Health Inform.* 2017, 21, 4–21. [CrossRef]
14. Hannun, A.Y.; Rajpurkar, P.; Haghpanahi, M.; Tison, G.H.; Bourn, C.; Turakhia, M.P.; Ng, A.Y. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat. Med.* 2019, 25, 65. [CrossRef] [PubMed]
15. Chauhan, S.; Vig, L. Anomaly detection in ECG time signals via deep long short-term memory networks. In *Proceedings of the IEEE International Conference on Data Science & Advanced Analytics*, Paris, France, 19–21 October 2015.
16. Page, A.; Kulkarni, A.; Mohsenin, T. Utilizing deep neural nets for an embedded ECG-based biometric authentication system. In *Proceedings of the Biomedical Circuits & Systems Conference*, Atlanta, GA, USA, 22–24 October 2015.
17. Zühlmann, M.; Perekrestenko, D.; Tschannen, M. Convolutional Recurrent Neural Networks for Electrocardiogram Classification. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017; pp. 124–132.
18. Narayan, S.M.; Rodrigo, M.; Kowalewski, C.A.; Shenasa, F.; Meckler, G.L.; Vishwanathan, M.N.; Wang, P.J. Ablation of Focal Impulses and Rotational Sources: What Can Be Learned from Differing Procedural Outcomes? *Curr. Cardiol. Risk Rep.* 2017, 11, 27. [CrossRef]
19. Wang, R.; Fan, J.; Li, Y. Deep Multi-Scale Fusion Neural Network for Multi-Class Arrhythmia Detection. *IEEE J. Biomed. Health Inform.* 2020, 24, 2461–2472. [CrossRef]
20. Zhang, J.; Liu, A.; Gao, M.; Chen, X.; Zhang, X.; Chen, X. ECG-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network. *Artif. Intell. Med.* 2020, 106, 101856. [CrossRef] [PubMed]

21. Mittal, G.; Yagnik, K.B.; Garg, M.; Krishnan, N.C. SpotGarbage: Smartphone App to Detect Garbage Using Deep Learning. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp’16, Heidelberg, Germany, 12–16 September 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 940–945. [CrossRef]

22. Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Meghini, C.; Vairo, C. Deep learning for decentralized parking lot occupancy detection. *Expert Syst. Appl.* 2016, 72, 327–334. [CrossRef]

23. Ravi, D.; Wong, C.; Lo, B.; Yang, G.Z. A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices. *IEEE J. Biomed. Health Inform.* 2017, 21, 56–64. [CrossRef]

24. Nguyen, V.T.; Nguyen-Thanh, N.; Yang, L.; Nguyen, D.H.N.; Jabbour, C. Cognitive computation and communication: A complement solution to cloud for IoT. In Proceedings of the International Conference on Advanced Technologies for Communications, Hanoi, Vietnam, 12–14 October 2016.

25. Zhang, K.; Xu, G.; Han, Z.; Ma, K.; Zheng, X.; Chen, L.; Duan, N.; Zhang, S. Data Augmentation for Motor Imagery Signal Classification Based on a Hybrid Neural Network. *Sensors* 2020, 20, 4485. [CrossRef]

26. Zanini, R.; Colombini, E. Parkinson’s Disease EMG Data Augmentation and Simulation with DCGANs and Style Transfer. *Sensors* 2020, 20, 2605. [CrossRef]

27. Cao, P.; Li, X.; Mao, K.; Lu, F.; Pan, Q. A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation. *Biomed. Signal Process. Control* 2020, 56, 101675. [CrossRef]

28. Salamon, J.; Bello, J.P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Process. Lett.* 2017, 24, 279–283. [CrossRef]

29. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* 2017, arXiv:1712.04621.

30. Lane, N.D.; Bhattacharya, S.; Georgiev, P.; Forlivesi, C.; Kawser, F. An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-of-Things Devices. In Proceedings of the 2015 International Workshop on Internet of Things towards Applications, IoT-App’15, New York, NY, USA, 1 November 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 7–12. [CrossRef]

31. Mehdi, M.; Ala, A.F.; Sameh, S.; Mohsen, G. Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Commun. Surv. Tutor.* 2018, 20, 2923–2960.

32. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* 2015, arXiv:1502.03167.

33. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 630–645.

34. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Montreal, QC, Canada, 2014; pp. 2672–2680.

35. Hedén, B.; Ohlsson, M.; Holst, H.; Mjöman, M.; Rittner, R.; Pahlm, O.; Peterson, C.; Edenbrandt, L. Detection of frequently overlooked electrocardiographic lead reversals using artificial neural networks. *Am. J. Cardiol.* 1996, 78, 600–604. [CrossRef]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).