ICU false alarm identification based on convolution neural network

Qiang Yu, Cheng Wang, Jing Xi, Ying Chen, Weifeng Li, Yun Ge and Xiaolin Huang

Abstract

Background: In intensive care unit (ICU), excessive false alarms burden medical staff greatly, and cause medical resource waste as well. In order to alleviate false alarms in ICU, we constructed models for classification using convolutional neural networks, which can deal directly with time series and avoid extracting features manually.

Results: Combining with grouping strategy, we tried two basic network structures, i.e. DGCN and EDGCN. After that, based on EDGCN, which was proved better, ensembling networks were also constructed to elevate the performance further. Considering of the limited sample size, different data expansions were also experimented. Finally, we tested our model in the online sandbox, and got a score of 78.14.

Conclusions: Although the performance is slightly lower than the best scores that have been reported, our models are end-to-end, through which the original time series can be automatically mapped into a binary output, without manually feature extraction. In addition, our method innovatively uses grouped convolution to make full use of the information in multi-channel signals. In the end, we also discussed the potential solutions to further elevate performances.

Keywords: ICU; DGCN; EDGCN; grouped convolution; multi-channel physiological signals

1 Introduction

In the intensive care unit (ICU), the alarm issued by monitoring equipments can promptly notify the medical staff of patients’ abnormal condition, and protect patients from danger. However, these automatic alarms are designed for high sensitivity rather than specificity, leading to excessive false alarms. According to previous research, nearly 88.8% of alarms in ICU are false[1] and false arrhythmia alarms are commonly due to single channel ECG artifacts and low voltage signals[2]. Such frequent false alarms will inevitably burden medical staff greatly, and the noise generated by the alarm will also stimulate the patient, resulting in patient delirium[3]. Therefore, it is quite necessary to examine the ICU alarms carefully and reduce the false alarm as far as possible.

In 2015, PhysioNet launched a competition of reducing false arrhythmia alarms in ICU, and after that, more researchers have devoted to this area. Researchers have tried many ways varing from dedicated decision rules to non-dedicated machine learning.

The rule-based decision-making system focuses on how to use multi-modal signals to estimate heart rate more accurately. In the decision-making process, according to
the characteristics of the signal, many heart rate estimating algorithms and decision rules are fine-tuned. These improvements include invalid data or noise detection\cite{6,13,15}, signal filtering\cite{7}, signal quality indices(SQIs) or assessment(SQA)\cite{7,8,9,10}, pulse detection\cite{7,11}, QRS detection\cite{4,6,9,10,12,13}, spectral features\cite{8}, heart-beat estimation\cite{4,6,9}, and multi-rules fusion\cite{4,6,10,12}, etc. For example, in the challenge, Plesinger and colleagues\cite{6} established a model based on a series of decision rules and obtained a final score of 81.39 on the hidden testing set of the competition and ranked 1st in the competition. Krasteva and colleagues\cite{13} first used ADLib(Schiller AG) and Pulse Wave Analysis Module(PWAM) to analyze ECG, arterial blood pressure(ABP) and photoplethysmogram(PPG) signals(if available) respectively. Then, based on the extracted characteristics, they constructed a rule-based decision model to do the judgement. They got 79.49 score in the final hidden set test.

Machine learning has gained much attention these years. However, due to the nature of regular monitoring physiological signals including heterogeneity, low signal to noise ratio, high deficiency and high temporal resolution over long periods of time\cite{14}, applying machine-learning approaches to them is quite challenging, and much human intervention is required. Machine learning algorithms usually focus on extracting features that can distinguish different alarms, and use algorithms such as majority voting\cite{15,16}, SVM\cite{17,18}, decision trees\cite{19}, random forests\cite{20,21,22} for classification. Most machine-learning algorithms use the same inputting parameters as the rule-based strategies, such as SQIs\cite{19,20,22}, heartrate\cite{15,19,22}, QRS detection\cite{17}, etc. In addition to basic statistical and spectral features, certain derived features designed according to the alarm type are also used\cite{20,22}. Due to the large number of features, machine learning solutions often require feature dimensionality reduction, such as using PCA\cite{15,23} and feature selections\cite{22}. For example, considering that false alarms were usually caused by low QRS complex amplitude and high noise, Au-Yeung and colleagues\cite{22} first processed the signal to ensure that the heartbeat can be correctly labeled and then extracted features from the signal. In particular, they extracted a series of SQIs to distinguish noise or artifacts from real physiological signals. After that, they trained a random forest model using all extracted features. Their model got a 83.08 score on the hidden set.

In another way, Hooman and colleagues\cite{24} have proposed a neural network training method based on genetic evolution algorithm. They used the features extracted by Liu\cite{4} for network training. According to the cross-validation on the open dataset, their methods obtained the highest score of 86.81. Gajowniczek and colleagues\cite{21} have proposed a weight-based random forest method. They reported that their model performed better than standard random forest, adaboost and other ensemble models on the open dataset.

Deep learning has also been applied to this field. Schwab and colleagues\cite{14} has proposed Distantly Supervised Multitask Networks to reduce false alarms. Their innovative neural network includes multiple perceptual blocks, multiple auxiliary task models, combined with a final classification unit, and they trained auxiliary task and final classification unit alternately. They reported a better result than those of other methods on their own datasets. Nevertheless, in their method, extracting and selecting features manually are still necessary. Lehman and colleagues\cite{25} firstly
performed FFT on ECG, and then used an auto-encoder to reconstruct the signal. The low-dimensional expression derived from the encoder was used as the input feature of the neural network for classification. Focusing on the ventricular tachycardia alarm (VTA), their method has a performance improvement of nearly 1% compared to Plesinger’s [6] on their own validation set.

In general, above methods greatly relied on manually extracted features. Although, Lehman and colleagues [25] got rid of part of the feature extraction work through the auto-encoder, they still performed lots of signal preprocessing, such as FFT, peak detection, etc. Breaking through the limitation of manually extracting features, we attempted to accomplish features extraction and final classification all by automation. As former walkers, Schirrmeister and colleagues [26] successfully designed a deep convolutional neural network (CNN) with raw EEG signals inputting and classified labels outputting. Their method performs the same as or even better than traditional methods using pre-extracted spectrum features. Besides, the features used are not fixed a priori.

Building on their basic CNN ideas, we designed a deep grouped convolutional neural network (DGCN) for the ICU false alarm task. Main difference from Schirrmeister’s network lies in that we took the grouping strategy instead of simple two-dimension convolution to ensure that information from multi-channel heterogeneous signals, possibly of different intrinsic time scales, can be extracted and reserved respectively for final classification. That is, the multi-channel raw signals are directly inputted into the network and convolved seperately by different kernels according to their signal type. The features automatically extracted from different channels are all reserved until they are fused at the end of the network to complete the classification of true or false alarms. Grouping first combined with fusion last achieves an information complement from different channels. Based on the DGCN, the embedded deep group convolutional network (EDGCN) has also been developped. We compared above two networks and constructed ensemble models using EDGCN architecture. The score of our best performance model is 78.17 on the hidden set of 2015 physionet competition.

2 Data and Methods

2.1 Data Descriptions

We used the database of 2015 physionet/CinC challenge, i.e. reducing false arrhythmia alarms in the ICU. It includes a total of 1,250 arrhythmia alarm records, classified as one of five arrhythmia types of asystole (ASY), extreme bradycardia (EBR), extreme tachycardia (ETC), ventricular tachycardia (VTA) and ventricular fibrillation or flutter (VFB) [3]. Each alarm was labelled by ‘True’ or ‘False’ confirmed by clinical experts. However, only 750 records are open access. The remained 500 were embedded in a sandbox environment for testing services provided by the challenge organizers [3]. Therefore, we used the available 750 records for modeling as well as cross-validation, and tested our model in the sandbox environment. The basic statistical information about the open-access 750 records is shown as Table 1.

Each record contains three or four synchronized physiological time series, including two leads of ECG and one or two supplementary signals as arterial blood pressure (ABP), photoplethysmogram (PPG) or respiration (RESP) [3]. The number of
Table 1 Basic statistical information of 750 open records[3, 13]

| Arrhythmia | Count/Ratio | Arrhythmia Symptoms | True | False |
|------------|-------------|---------------------|------|-------|
| ASY        | 122/17%     | 0 beats in 4s       | 22   | 100   |
| EBR        | 89/11%      | ≥5 beats, HR < 40 bpm | 43   | 46    |
| ETC        | 140/17%     | ≥17 beats, HR > 140 bpm | 131  | 9     |
| VTA        | 341/47%     | ≥5 ventr.beats, HR > 100 bpm | 89   | 252   |
| VFB        | 58/7%       | Fibrillation waves in 4s | 6    | 52    |

According to the data description from the challenge organizer, all these time series have been resampled to 12 bit, 250Hz and band- as well as notch- filtered, therefore, we did not take extra filtering. As to the series length, 375 records are of 5 minutes length, and 375 are of 5.5 minutes. As told by the organizer, all the alarm occurs at 5:00 and the suspected arrhythmia event triggering the alarm is believed occurring between 4:50 to 5:00[27]. Considering that there might have been additional unannotated arrhythmia events in the 5 minutes preceding the alarm, we only use 15s duration from 4:45 to 5:00, in order to eliminate the uncertainty of unmarked series duration as well as retain information as much as possible.

2.2 Two basic Network Structures

We tried two networks. One is a deep CNN combined with grouping strategy(DGCN). The other employs an extra embedding layer to introduce arrhythmia type information, and we name it as embedded deep group convolutional network(EDGCN).

2.2.1 DGCN Net

The DGCN model adopts a five-layer structure as shown in Fig. 1. As shown, layer 1-4 are of basically the same structure, which sequentially comprises a convolution stage, a batch normalization[28] stage, a Leaky Relu activation[29, 30] stage, and a maximum pooling[31] stage.

Considering of the different physiological mechanisms and temporal scales underlying different signals, i.e. ECG, blood pressure and respiration, we use a grouped convolution strategy in order to extract the specific characteristics of corresponding signal. That is, as shown in Fig. 1, the different input series are treated as different groups, which are marked by different colors, i.e. red, green or blue. Each group, represented by retaining its original color, will not mix up till the last stage. To be specific, in each layer from layer 1 to 4, each group is independently convolved by 15(layer 1 and 2) or 30 (layer 3 and 4) different kernels of stride 1, length 10, width $K_i$, and no padding, in which $K_i$ represents the number of channels in each group.
of the $i - th$ layer input. Thus for $N$ input series or groups, there are $N \times 15$ (or $N \times 30$) different two-dimension kernels, generating $N \times 15$ (or $N \times 30$) different convolved series. Then, batch normalization [28] is conducted to alleviate the disappearance of the gradient and speed up the model training. The following is Leaky Relu activation stage. Finally, we use max pooling of width 3 and stride 3, which means downsampling ratio 3. In each layer, each neuron’s receptive field, in length of original series, is illustrated as Fig. 2. In addition, we also depicted the length of output of each layer.

Layer 5 is the decision layer, which comprises a convolution, an exponential linear unit (ELU) activation function [32], a full-connected (FC) layer (via convolution), and
softmax in order. In order to integrate the information extracted from different groups, we first use $N$ convolution kernels of stride 1, length 3, width $K_5 \times N$ to combine the different group results of the preceding forth layer, and then use the ELU activation function to perform nonlinear transformation. After that, we use 2 convolution kernels whose size are the same as the output of the previous step to perform convolution, as a FC layer, and get a result vector of dimension two. Dropout can effectively alleviate overfitting, therefore, we adopted it in FC layer and the dropout ratio is set 0.3. Finally, we transform our result into a decision probability through Softmax.

2.2.2 EDGCN

EDGCN is mostly the same as DGCN, except that in layer 5 we embed an external vector, which is the the one-hot coding of the arrhythmia symptom for each alarm, into the model. To be specific, the $N \times 30$ output series of length $L_9$ from layer 4 were first group-convolved with $N$ corresponding kernels of stride 1, length 3 and width $30 \times N$, after which we get $N$ series of length $L_{10} = L_9 - 2$. Following ELU stage, we reshape the outputing matrix of $N$ rows and $L_{10}$ columns into a vector. Then we concat the one-hot coding of arrhythmia type, of length 5, with the reshaped vector to obtain a vector of length $N \times L_{10} + 5$. After that, there is another convolution stage with 32 kernels of length $N \times L_{10} + 5$, working as a FC layer. And finally, we use another FC layer to compress the 32-dimension output vector from the previous stage into a 2-dimension vector and use a softmax classification to get the final results. Dropout is also adotpted in the FC layer, and the dropout ratio is set 0.3 The block chart of layer 5 as well as the data shape in each stage is illustrated in Fig. 3.

![Layer 5 block diagram](image_url)
3 Experiments and results

3.1 Data Preprocessing

In data preprocessing stage, we simply did necessary NAN filling. We adopted forward-filling. That is, for missing values, we first examine the preceding point. If it is a valid value, we fill the missing with the preceding value; whereas it is not, which happens irregularly, we simply retain the NAN.

3.2 Data expansion

Noting that the sample size of training set (750) might be insufficient, we made a data expansion by sliding window of length 12s and step 0.5s. Thus, for each 15s original series, we obtained 7 crops (or slices) and expanded the training set 7 times as the original. As to the application of the trained model, we also used the same cropping strategy, and merged all 7 crops results by voting.

Then, considering that the expanded data has considerable redundancy, we randomly add artificial noises generated via formula (1) to each slice.

\[ x_{i}^{noise} = x_i + A\epsilon \]
\[ = x_i + \frac{x_{max} - x_{min}}{10} \epsilon, \]
\[ \epsilon \sim N(0,1), i = 1, 2, \ldots , 250 \times 12 \]

Figure 4 Schematic diagram of slicing process.
in which $x_{\min} = \min\{x_i|i = 1, 2, \cdots, 250 \times 12\}$ and $x_{\max} = \max\{x_j|j = 1, 2, \cdots, 250 \times 12\}$.

In formula (1), $x^{\text{noise}}$ represents the slice after adding noise, $\epsilon$ represents random noise obeying standard normal distribution. $A$ is the amplitude coefficient of the added noise and was set as 10% of the original signal peak-to-peak amplitude.

Waveforms of a representative series before and after noise addition are shown as Fig. 5.

In certain experiments, we also tried 0-1 normalization in each slice as formula (2).

$$\hat{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, i = 1, 2, \cdots, 250 \times 12$$ (2)

in which $x_{\min} = \min\{x_i|i = 1, 2, \cdots, 250 \times 12\}$ and $x_{\max} = \max\{x_j|j = 1, 2, \cdots, 250 \times 12\}$.

3.3 Performance Evaluation

Besides the frequently used measures as accuracy(ACC), area under the receiver characteristic operator curve(AUC-ROC), true positive rate(TPR) and true negative rate(TNR), we also adopt the compound Score measure designed by Clifford and colleagues[3] as the evaluation index of the model. As shown by the definition formula (3), Score index punishes the false negative(FN) considering that it is quite dangerous to report a true alarm as false.

$$Score = \frac{100 \cdot (TP + TN)}{TP + TN + FP + 5 \times FN}$$ (3)
3.4 Experiments
We experimented with the two network structures introduced in Section 2.2. In both networks, we selected the II, V, ABP, RESP, and PPG as the model input. We adopted the logistic loss, and used the Adam optimizer[33]. The hyperparameters, which were exactly the same in DGCN and EDGCN, are listed as Table 3.

| Model  | Dropout(%) | learning rate | batch size | early stop | max epoch | noising probability |
|--------|------------|----------------|-------------|------------|------------|---------------------|
| DGCN/EDGCN | 30%        | 0.0001         | 64          | 80         | 400        | 0.8                 |

We also compared 4 different data expansions, i.e. pure 12s slicing signals and 12s slicing with either noise addition or 0-1 normalization, or both with complete 15s signal without expansions.

In machine learning, model ensembling usually achieves better performance. Therefore, we also tried combination models illustrated as Fig. 6. It means that we trained independently four sub-models (in DGCN or EDGCN structures) using different input signals, and then combined them by averaging their results and comparing it with a threshold to determine a binary output.

![Diagram](image)

Figure 6 Schematic diagram of the combined model.

The overall execution flow of our experiments is shown in Fig. 7. All experiments ran in a 5-fold cross-validation way.

3.5 Results
Fig. 8 gives the profiles of DGCN, EDGCN and combined models 5-fold cross validation performances, i.e. ACC, AUC-ROC, and Scroe, on the open dataset. Different colors represent different data expansions. From Fig. 8, it can be seen that
Figure 7 Framework diagram of overall processing flow.

Figure 8 The first 200 epochs performances of different models on the open dataset.
for almost all occasions, models with data expansions can stabilize much quicker in the early 50 epoches. It means that 12s slicing does help accelerate training via increasing the amount of training data. Meanwhile, models of 12s slicing without 0-1 normalization perform better than both complete 15s signals without expansion and 12s slicing with 0-1 normalization. However, after 12s slicing, noise addition itself does not bring about obvious elevation. The possible reason lies in that interruptions in real environment are usually not Gaussian stochastic.

In order to make a clearer comparison between models, we list in Table 4 the best performance measures for each model. From Table 4, we can find that on the open dataset, combined EDGCN models have the highest scores, and following are EDGCN models. DGCN models perform the most poorly. It implies that the additional arrhythmia type embedding in EDGCN does supply information helpful for classification. What is more, ensembling strategy can improve performance further.

For the same network architecture, models using 12s slicing without 0-1 normalization perform the best, and then rank models using 12s slicing with 0-1 normalization. Models using complete 15s signals perform the worst. On the other side, noise addition only slightly elevate the performance compared with using pure 12s slicing.

Table 4 Performance of different experiments on open dataset

| Model Descriptions                       | TP  | FP  | TN  | FN  | ACC  | TPR  | TNR  | Score  |
|------------------------------------------|-----|-----|-----|-----|------|------|------|--------|
| DGCN 15s record                          | 257 | 74  | 382 | 37  | 85.20| 87.41| 83.77| 71.15  |
| DGCN 12s slice                           | 269 | 103 | 353 | 25  | 82.93| 91.49| 77.41| 73.17  |
| DGCN 12s slice, 0-1 normalization        | 258 | 111 | 345 | 36  | 80.40| 87.75| 75.65| 67.44  |
| DGCN 12s slice, noise addition           | 267 | 88  | 368 | 27  | 84.46| 90.81| 80.70| 74.01  |
| DGCN 12s slice, 0-1 normalization, noise addition | 262 | 110 | 346 | 32  | 81.06| 89.11| 75.87| 69.24  |
| EDGCN 15s record                         | 268 | 83  | 373 | 26  | 85.46| 91.15| 81.79| 75.05  |
| EDGCN 12s slice                          | 272 | 86  | 370 | 22  | 85.60| 92.51| 81.14| 76.61  |
| EDGCN 12s slice, 0-1 normalization       | 260 | 72  | 384 | 34  | 85.86| 88.43| 84.21| 72.68  |
| EDGCN 12s slice, noise addition          | 270 | 85  | 371 | 24  | 85.46| 91.83| 81.35| 75.76  |
| EDGCN 12s slice, 0-1 normalization, noise addition | 267 | 111 | 345 | 27  | 81.60| 90.81| 75.65| 71.32  |
| Combined EDGCN, 15s record               | 265 | 67  | 389 | 29  | 87.20| 90.13| 85.30| 75.51  |
| Combined EDGCN, 12s slice                | 275 | 67  | 389 | 19  | 88.53| 93.53| 85.30| 80.38  |
| Combined EDGCN, 12s slice, 0-1 normalization | 273 | 90  | 366 | 21  | 85.20| 92.85| 80.26| 76.61  |
| Combined EDGCN, 12s slice, noise addition | 277 | 71  | 385 | 17  | 88.26| 94.21| 84.42| 80.92  |
| Combined EDGCN, 12s slice, 0-1 normaliza-|    |    |    |    | 85.86| 93.87| 80.70| 78.34  |
| tion, noise addition                     |    |    |    |    |      |      |      |        |

1 Bold represents the best performance in the current experimental scheme.

Finally we tested our combined models in the official sandbox environment. Performance measures are listed in Table 5. As we see, the combined EDGCN model using 12s slicing and normalization got the best score of 78.17. It is interesting that on the hidden set of sandbox environment, data expansion with 0-1 normalization achieves better performance, contrasting to the results on open dataset. One potential reason may be that the hidden dataset is different in amplitude from the open dataset, might resulting from different sampling system. Normalization in slicing can exactly erase this difference in absolute amplitude while retaining the relative shape information of the waveform.

For comparison, in Table 5 we also list Plesinger’s results, which ranked the first in the challenge. We can see that our best score is still slightly lower than Plesinger’s score. Compared with Plesinger’s model, the poor performances on the ASY alarm,
no higher than 75 contrasting to 97, underlies the main deficiency. As definition, ASY means no beats in 4s. It is not difficult for a rule-based model to detect the real ASY. However, in our models, as our kernel length determined, the times scale reaches approximately 2.4s till the last convolution. And then the final FC stage just compress all features in time scale no longer than 2.4s together. It means that actually we cannot comprehensively examine 4s continuous series. That may be one of the main reasons that all our models perform much worse than Plesinger’s on ASY.

Table 5 Combined Models’ performance on the hidden set

| Model           | Descriptions                  | Arrhythmia type | Asystole | TPR | TNR | Score |
|-----------------|-------------------------------|-----------------|----------|-----|-----|-------|
| Combined        | EDGCN 15s record              | ASY             | 78%      | 82% | 73.68 |
|                 |                               | EBR             | 54%      | 76% | 38.46 |
|                 |                               | ETC             | 100%     | 60% | 98.20 |
|                 |                               | VFB             | 100%     | 92% | 93.10 |
|                 |                               | VTA             | 54%      | 87% | 58.28 |
|                 | Total                         |                 | 78%      | 84% | 64.22 |
| Combined        | EDGCN 12s slice               | ASY             | 78%      | 82% | 73.68 |
|                 |                               | EBR             | 92%      | 72% | 71.56 |
|                 |                               | ETC             | 100%     | 0%  | 95.50 |
|                 |                               | VFB             | 89%      | 100%| 91.94 |
|                 |                               | VTA             | 72%      | 82% | 65.19 |
|                 | Total                         |                 | 89%      | 82% | 74.12 |
| Combined        | EDGCN 12s slice 0-1 normalization | ASY           | 78%      | 79% | 71.35 |
|                 |                               | EBR             | 100%     | 71% | 82.47 |
|                 |                               | ETC             | 99%      | 20% | 92.17 |
|                 |                               | VFB             | 78%      | 90% | 77.27 |
|                 |                               | VTA             | 90%      | 80% | 75.91 |
|                 | Total                         |                 | 94%      | 79% | 78.17 |
| Combined        | EDGCN 12s slice noise addition | ASY             | 78%      | 81% | 73.10 |
|                 |                               | EBR             | 92%      | 72% | 71.56 |
|                 |                               | ETC             | 100%     | 0%  | 95.50 |
|                 |                               | VFB             | 89%      | 100%| 91.94 |
|                 |                               | VTA             | 72%      | 80% | 63.95 |
|                 | Total                         |                 | 89%      | 81% | 73.42 |
| Combined        | EDGCN 12s slice 0-1 normalization noise addition | ASY           | 78%      | 72% | 66.08 |
|                 |                               | EBR             | 100%     | 62% | 77.32 |
|                 |                               | ETC             | 99%      | 20% | 92.17 |
|                 |                               | VFB             | 100%     | 86% | 87.93 |
|                 |                               | VTA             | 90%      | 73% | 70.59 |
|                 | Total                         |                 | 96%      | 73% | 75.57 |
| Plesinger, et. [6] | Challenge Winner              | ASY             | 100%     | 97% | 97.42 |
|                 |                               | EBR             | 100%     | 72% | 83.51 |
|                 |                               | ETC             | 97%      | 100%| 87.80 |
|                 |                               | VFB             | 67%      | 100%| 78.57 |
|                 |                               | VTA             | 85%      | 84% | 75.07 |
|                 | Total                         |                 | 92%      | 88% | 81.39 |

1 Bold represents the best performance in the current experimental scheme.

4 Conclusion and Discussion

In this work, we developed an end-to-end model for ICU false alarm identification, through which the original time series can be automatically mapped into a binary output. One of the most principle advantage of this kind of methods is that manually extracting features is not pre-requisite. Convolution, which acts actually as filtering, conducts the feature extraction by itself. In addition, by grouping strategy, we make full use of various synchronized time series that represent different physiological aspects.
However, as shown by results, the score on the official hidden set is not that good, although it is not too bad. We reflected it and believe that the reasons as well as the potential solutions might lie in the following aspects:

Firstly, in this work, although we adopted group convolution strategy in order to carefully retain the characteristics of different series types, we did not truly adopt different receptive fields or time scales for them. As we know, ECG has much higher frequency than respiration, and PPG as well as ABP is great different from ECG in waveform, which implies the typical temporal response differences between electrophysiology and hemodynamics. However, we only adopted one set of kernel lengths for all different series types, especially in the first convolution stage that determines the basic coarse-grained scale. As Fig. 2 shows, in the first convolution stage, the receptive field is set 0.04s. It is a trade-off between ‘fast’ ECG and ‘slow’ RESP. The direct consequence is that it may be exactly appropriate for neither. Therefore, in order to promote the model performance, designing different kernel length for different series type is a potential solution.

Secondly, as arrhythmia type varies, the requisite or shortest continuous series length also varies. However, our methods did not make corresponding variation to satisfy the different arrhythmia type. For example, as discussed in the results, for ASY the model lacks a comprehensive scale of 4s. Therefore, in future work, we should design the kernel length more particularly for particular arrhythmia type.

Thirdly, in order to relieve the problem brought about by deficient samples, we adopted data expansion via sliding a 12s-window with stride 0.5s from 15 seconds previous alarming to exactly alarm happening. This conduction on the one hand does help elevate the model performance, which can be inferred from Fig. 8 and Table. 4; on the other hand, it may introduce slices with mistaken labels, e.g. slices labelling "True" alarm happens to embrace no arrhythmia events. This occasion happens especially when arrhythmia event can quickly trigger alarming with delay no longer than 5 seconds. Unfortunately, this problem can not be solved by simply increasing or decreasing the window length. Instead of data expansion, getting more reliable real samples is the final solution.

Acknowledgements
The authors thank for physionet challenge 2015 organizers provided ICU dataset and sandbox test environment for research.

Funding
Natural Science Foundation of Jiangsu Province(BK20191250, BK20190309), Social Development Program of Primary Research & Development Plan in Jiangsu Province(BE2020685, BE2017679), foundation for the cooperation between school and enterprise of Nanjing City(201722002).

Availability of data and materials
Our experimental code will be available at: here. In addition, the code needs our authorization to use it publicly.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Authors’ contributions
Q.Yu and X.Huang designed the experiment, network structure and drafted the initial version of the manuscript, and X.Huang reviewed and revised it. Q.Yu and C.Wang conducted a lot of experiments to verify the results. J.Xi provided help for the experiment. Y.Chen, W.Li, Y.Ge participated in the revision of the manuscript. All authors read and approved the final manuscript.
Author details
School of Electronic Science and Engineering, Nanjing University, Nanjing, China.

References

1. Drew, B.J., Harris, P., Zegre-Hemsey, J.K., Mammane, T., Schindler, D., Salas-Boni, R., Bai, Y., Tinoco, A., Ding, Q., Hu, X.: Insights into the problem of alarm fatigue with physiologic monitor devices: A comprehensive observational study of consecutive intensive care unit patients. PLoS ONE 9(10) (2014)
2. Aboukhalil, A., Nielsen, L., Saeed, M., Mark, R.G., Clifford, G.D.: Reducing false alarm rates for critical arrhythmias using the arterial blood pressure waveform. Journal of Biomedical Informatics 41(3), 442–451 (2008)
3. Clifford, G.D., Silva, I., Moody, B., Li, Q., Kella, D., Shahin, A., Kooiistra, T., Perry, D., Mark, R.G.: The PhysioNet/Computing in Cardiology Challenge 2015: Reducing false arrhythmia alarms in the ICU. Computing in Cardiology 42, 273–276 (2015)
4. Liu, C., Zhao, L., Tang, H.: Reduction of False Alarms in Intensive Care Unit using Multi-feature Fusion Method. Computing in Cardiology 42(2), 741–744 (2015)
5. Tsimenidis, C., Murray, A.: Reliability of clinical alarm detection in intensive care units. Computing in Cardiology 42, 1185–1188 (2015)
6. Plesinger, F., Klimes, P., Halamek, J., Jurak, P.: False alarms in intensive care unit monitors: Detection of life-threatening arrhythmias using elementary algebra, descriptive statistics and fuzzy logic. Computing in Cardiology 42, 281–284 (2015)
7. Zong, W.: Reduction of false critical ECG alarms using waveform features of arterial blood pressure and/or photoplethysmogram signals. Computing in Cardiology 42, 289–292 (2015)
8. Fallet, S., Yazdani, S., Vesin, J.-M.: A multimodal approach to reduce false arrhythmia alarms in the intensive care unit, 277–280 (2015)
9. Couto, P., Ramalho, R., Rodrigues, R.: Suppression of false arrhythmia alarms using ECG and pulsatile waveforms. Computing in Cardiology 42, 749–752 (2015)
10. Sadr, N., Huvanandana, J., Nguyen, D.T., Kalra, C., McEwan, A., De Chazal, P.: Reducing false arrhythmia alarms in the ICU by Hilbert QRS detection. Computing in Cardiology 42, 1173–1176 (2015)
11. Ansari, S., Belle, A., Najarian, K.: Multi-modal integrated approach towards reducing false arrhythmia alarms during continuous patient monitoring: The Physionet Challenge 2015. Computing in Cardiology 42, 1181–1184 (2015)
12. Serackis, A., Abromavičius, V., Guddiešis, A.: Identification of ECG signal pattern changes to reduce the incidence of Ventricular Tachycardia false alarms. Computing in Cardiology 42, 1193–1196 (2015)
13. Krasteva, V., Jekova, I., Leber, R., Schmid, R., Abächerli, R.: Validation of arrhythmia detection library on bedside monitor data for triggering alarms in intensive care. Computing in Cardiology 42, 737–740 (2015)
14. Schwab, P., Keller, E., Mursi, C., Mack, D.J., Sträßle, C., Karten, W.: Not to cry Wolf: Distantly supervised multistask learning in critical care. 35th International Conference on Machine Learning, IML 2018 10, 7185–7195 (2018). 1802.05027
15. Hoog Antink, C., Leonhardt, S.: Reducing false arrhythmia alarms using robust interval estimation and machine learning. Computing in Cardiology 42(5), 285–288 (2015)
16. Teo, S.-k., Wong, J.C., Yang, B., Yang, F., Feng, L., Lim, T.W., Su, Y.: Reducing False Arrhythmia Alarms in the ICU Institute of High Performance Computing, A * STAR, Singapore. Computing in Cardiology 42, 1177–1180 (2015)
17. Kalidas, V., Tamil, L.S.: Enhancing accuracy of arrhythmia classification by combining logical and machine learning techniques. Computing in Cardiology 42, 733–736 (2015)
18. Daluwatte, C., Johannesen, L., Vicente, J., Scully, C.G., Galeotti, L., Strauss, D.G.: Heartbeat fusion algorithm to reduce false alarms for arrhythmias. Computing in Cardiology 42, 745–748 (2015)
19. Caballero, M., Mirsky, G.M.: Reduction of false cardiac arrhythmia alarms through the use of machine learning techniques. Computing in Cardiology 42, 1169–1172 (2015)
20. Eerikäinen, L.M., Vanschooren, J., Rooijakkers, M.J., Vullings, R., Aarts, R.M.: Decreasing the false alarm rate of arrhythmias in intensive care using a machine learning approach. Computing in Cardiology 42, 293–296 (2015)
21. Gajowniczek, K., Grzegorczyk, I., Zabkowska, T., Bajaj, C.: Weighted random forests to improve arrhythmia classification. Electronics (Switzerland) 9(1), 1–20 (2020)
22. Au-Yeung, W.-T.M., Sahani, A.K., Iselbacher, E.M., Armoundas, A.A.: Reduction of false alarms in the intensive care unit using an optimized machine learning based approach. npj Digital Medicine 2(1), 1–6 (2019)
23. Shlens, J.: A tutorial on principal component analysis. CoRR abs/1404.1100 (2014). 1404.1100
24. Hooman Oroojeni, M.J., Majid Al-Rifaie, M., Nicolaou, M.A.: Deep neuroevolution: Training deep neural networks for false alarm detection in intensive care units. European Signal Processing Conference 2018-September(1), 1157–1161 (2018)
25. Lehman, E.P., Krishnan, R.G., Zhao, X., Mark, R.G., Lehman, L.-W.H.: Representation Learning Approaches to Detect False Arrhythmia Alarms from ECG Dynamics. Proceedings of Machine Learning Research 85, 1–15 (2018)
26. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., Ball, T.: Deep learning with convolutional neural networks for eeg decoding and visualization. Human Brain Mapping 38(11), 5391–5420 (2017)
27. American National Standard (ANSI/AAMI EC13:2002) Cardiac Monitors, Heart Rate Meters, and Alarms. VA: Association for the Advancement of Medical Instrumentation, Arlington (2002)
28. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167 (2015). 1502.03167
29. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. in ICML Workshop on Deep Learning for Audio, Speech and Language Processing 28 (2013)
30. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. CoRR abs/1505.00853 (2015). 1505.00853

31. Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6354 LNCS(PART 3), 92–101 (2010)

32. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs), 1–14 (2016). 1511.07289

33. Singarimbun, R.N., Nababan, E.B., Sitompul, O.S.: Adaptive moment estimation to minimize square error in backpropagation algorithm. In: 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), pp. 1–7 (2019)