S & M 3010

Arrhythmia Detection Using a Taguchi-based Convolutional Neuro-fuzzy Network

Jiarong Li,[1] Jyun-Yu Jhang,[2,3] Cheng-Jian Lin,[2,4*] and Xue-Qian Lin[4]

[1]College of Electrical Engineering, Yancheng Institute of Technology, Jiangsu 224000, China
[2]College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan
[3]Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology, Taichung 404, Taiwan
[4]Department of Computer Science and Information Engineering,
National Chin-Yi University of Technology, Taichung 411, Taiwan

(Received March 29, 2022; accepted May 23, 2022)

Keywords: arrhythmia detection, convolutional neural network, electrocardiography, neuro-fuzzy network, Taguchi method

With improvements in the quality of life, people have paid increased attention to their health. According to the American Heart Association, cardiovascular disease was one of the leading causes of death globally as of 2016. Medical experts estimate that the worldwide annual number of people dying from cardiovascular disease will reach 23.6 million by 2030. Detecting heart arrhythmias effectively and quickly is critical for preventing cardiovascular disease. In this paper, a one-dimensional Taguchi-based convolutional neuro-fuzzy network (1D-TCNFN) for detecting arrhythmia in electrocardiograms (ECGs) is proposed. The proposed 1D-TCNFN adopts neuro-fuzzy instead of conventionally connected layers to reduce the number of learned parameters in the network. Four feature fusion methods, namely, global average pooling, global max pooling, channel average pooling, and channel max pooling, are employed in the 1D-TCNFN. For an increased detection accuracy, the Taguchi method was used to optimize the network architecture of the proposed 1D-TCNFN. In the experiments, the open Massachusetts Institute of Technology–Beth Israel Hospital (MIT-BIH) Arrhythmia Database was adopted to verify the performance of the proposed method for detecting 17 different arrhythmia signals. The proposed 1D-TCNFN exhibited a detection accuracy of 93.95% for the MIT-BIH Arrhythmia Database.

1. Introduction

Arrhythmia is a common cardiovascular condition that is caused by abnormal electrical conduction in the heart. The most common clinical symptoms include palpitations, chest tightness, and chest pain.(1) According to the American Heart Association, as of 2016, cardiovascular disease caused 17.3 million deaths annually worldwide.(2) Electrocardiogram (ECG) analysis is typically used to determine the type of arrhythmia. Electrocardiogram signal measurement is a simple, fast, safe, and noninvasive procedure that can collect a large amount of

*Corresponding author: e-mail: cjlin@ncut.edu.tw
https://doi.org/10.18494/SAM3924

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

data for analysis in a short time. A standard ECG contains 12 recordings of current activity directions (six chest leads and six limb leads). Through the analysis of these 12 lead signals, heart diseases such as atrial fibrillation and myocardial infarction can be detected.(3) Arrhythmias can be either persistent or sporadic. Thus, for an accurate diagnosis, the ECG signals of some patients must be monitored continuously for an extended period (usually between 24 and 72 h). The collected ECG signal is then examined manually by a doctor. This method is time-consuming and inefficient, placing a substantial burden on medical personnel.(4)

To automatically detect abnormal ECG signals, many scholars have developed algorithms that analyze ECG signal characteristics. These algorithms are mainly based on the classification of Q, R, and S waves (QRS complexes) or the analysis of segmental ECG signals.(5,6) Alvarado *et al.*(7) proposed a time-based integrate-and-fire sampler encoding method that compresses the ECG signal and reduces the number of labeled heartbeats required by expert cardiologists. Mateo *et al.*(8) employed the radial basis function neural network to remove ectopic beats in ECG recordings to address mutual interference in ECG analysis. In the fragmented ECG signal analysis method, Lin and Zhang(9) proposed an autodetection algorithm that extracts features from ECG signals for classifying multiple types of cardiac states. The F1-score of the method reached 0.908, indicating that different types of cardiac abnormalities can be effectively detected. Padmavathi and Ramakrishna(10) compared the performance characteristics of k-nearest neighbors (KNN) and kernel support vector machine (SVM) by using 5, 15, and 30 s ECG segments. They found that the classification accuracy reached 90% when 30-s ECG segments were classified using the KNN with the Burg method. However, the classification of arrhythmia symptoms based on a QRS complex is highly challenging and error-prone. The features extracted from the ECG signal vary between patients. Therefore, arrhythmia detection methods using long-term ECG segments have increased in popularity.

Machine learning has been widely used in the medical field. Usman *et al.*(11) adopted machine learning for predicting epileptic seizures. First, the authors used empirical mode decomposition to extract features from electroencephalogram (EEG) signals. Subsequently, a prediction model was established by applying SVM. Thambiraj *et al.*(12) designed a system for predicting cardiovascular ailments. They employed a genetic algorithm (GA) to select physiological features and then used the random forest method to establish the prediction model. The experimental results indicate that the features selected by the GA can effectively improve the prediction accuracy of the model. Machine learning methods comprise the following three steps: signal feature extraction, feature selection, and predictive model construction.(13) The extraction and selection of features are critical for arrhythmia detection. Selecting appropriate features can help improve the accuracy of the prediction model. ECG features include heart rate variability, QRS complex width, and QT interval.(14) The above-mentioned literature points out that machine learning can be effectively applied in the medical field. However, these feature signals are extracted manually, making the consideration of all states practically impossible. Thus, deep learning methods with automatic feature extraction capabilities are suitable for signal analysis applications.(15–17) In this study, we employ a deep learning approach to solve the arrhythmia classification problem.

Among deep learning methods, the convolutional neural network (CNN) is the most widely known. Through the convolution operation, important features can be directly extracted from raw data instead of through manual extraction. The extracted features are then fed into fully connected layers to construct prediction or classification models. Although fully connected neural networks have excellent nonlinear mapping ability, they include 90% of the parameters and computations in the CNN. Effectively reducing the number of parameters in the network is the key for improving system performance. Numerous scholars have integrated fuzzy theory and neural networks, and have designed neuro-fuzzy networks such as ANFIS,(18) IT2FNN,(19) and FMM.(20) Unlike traditional neural networks, neuro-fuzzy networks combine fuzzy logic (similar to human reasoning) and the learning ability of the neural network, which can automatically construct rules and reduce the number of learnable parameters in the network. However, parameter design in the network architecture is also a key factor related to the overall performance. In engineering, the Taguchi method(21) is often used to optimize system parameters. In this method, the analysis of a small amount of experimental data can be used to effectively improve system performance.

In this study, we designed a one-dimensional (1D) Taguchi-based convolutional neuro-fuzzy network (1D-TCNFN), which combines the Taguchi method and a convolutional neuro-fuzzy network, for arrhythmia detection. The open Massachusetts Institute of Technology–Beth Israel Hospital (MIT-BIH) Arrhythmia Database(22) was used to experimentally evaluate the performance of the proposed 1D-TCNFN. The contributions of this study are as follows:

1. A new 1D-TCNFN for detecting arrhythmia is proposed.
2. Four feature fusion methods, namely, global average pooling, global max pooling, channel average pooling, and channel max pooling, are employed.
3. Compared with traditional CNN architectures, the proposed 1D-TCNFN has fewer network parameters and a higher detection accuracy.
4. The architecture of the 1D-TCNFN designed by employing the Taguchi method can be adjusted for different application problems.
5. The experimental results reveal that the proposed 1D-TCNFN can effectively detect different arrhythmia symptoms.

This paper is organized as follows. In Sect. 2, we describe the proposed method, in Sect. 3, we discuss the experimental results, and in Sect. 4, we provide the conclusions and future research directions.

2. Methods

In this section, we describe the proposed method for the detection of arrhythmia. First, we introduce the proposed 1D-TCNFN architecture in Sect. 2.1. Through the use of the 1D-TCNFN, 17 types of arrhythmia can be detected in long-term ECG signals (3600 sampling rate, equivalent to 10 s). We present the use of the Taguchi method for optimizing the hyperparameters of the 1D-TCNFN architecture in Sect. 2.2; in this manner, an optimized parameter combination can be obtained through a limited number of experiments.

2.1 Proposed 1D-TCNFN

The proposed 1D-TCNFN has a total of six layers comprising one input layer, three 1D convolutional layers, one feature fusion layer, one fuzzification layer, one rule layer, and one output layer. The 1D-TCNFN architecture is presented in Fig. 1.

The detailed description of each network layer in 1D-TCNFN is as follows.

(1) Input layer

X is a 1D signal with an input size of 3600 and is passed to the next layer.

$$X = \{x_1, x_2, \ldots, x_i\} \tag{1}$$

Here, i = 1, 2, …, 3600 is the index of the 1D signal.

(2) Convolutional layer

The 1D signal from the input layer is convolved with the convolution kernel to obtain different disease features. The operation of convolution is expressed as

$$y_{Conv}[f] = \sum_{k=1}^{ks-1} X[i+k+s] * w_{Conv}[f,k], \tag{2}$$

where y_{Conv} is the feature map output by the convolutional layer, f refers to the number of feature maps, k represents the index of the convolution kernel, ks is the size of the convolution kernel, s denotes the stride, and w_{Conv} is the convolution kernel. (Here, the Glorot uniform distribution is used to initialize the parameters of the convolution kernel.)

(3) Feature fusion layer

Choosing an appropriate feature fusion method in the feature fusion layer can reduce the complexity of the feature map and improve the model performance. Common feature fusion methods include global average pooling, global max pooling, channel average pooling, and channel max pooling, as presented in Fig. 2.

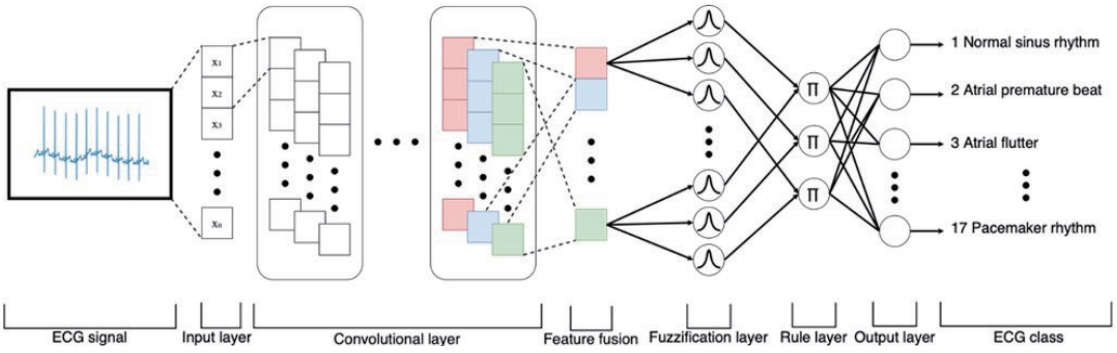


Fig. 1. (Color online) 1D-TCNFN architecture.

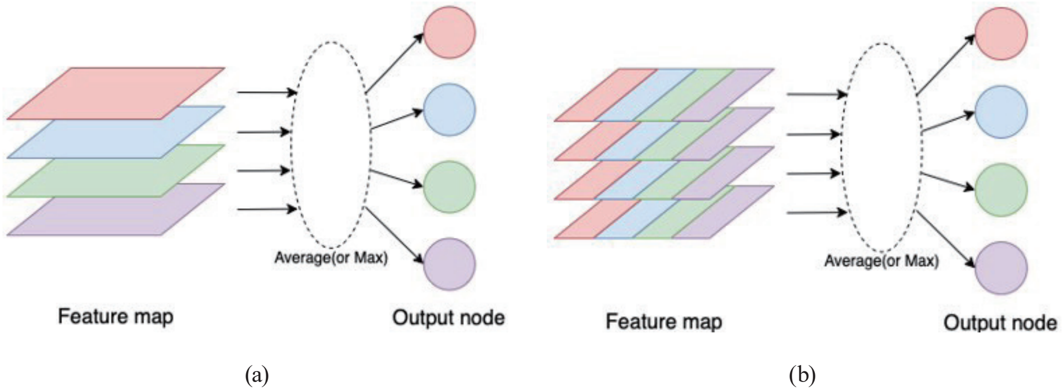


Fig. 2. (Color online) Feature fusion: (a) global pooling and (b) channel pooling.

$$y_{fusion}\begin{cases} y_{GAP}[f] = Avg\left(y_{Conv}[f]\right) \\ y_{GMP}[f] = Max\left(y_{Conv}[f]\right) \\ y_{CAP}[u] = Avg\left(y_{Conv}[u]\right) \\ y_{CMP}[u] = Max\left(y_{Conv}[u]\right) \end{cases} \tag{3}$$

Here, y_{GAP} represents the global average pooling that uses the average operation for each feature map, y_{GMP} indicates the global max pooling that uses the max operation for each feature map, y_{CAP} is the channel average pooling that uses the average operation for each channel of the feature map, and y_{CMP} denotes the channel max pooling that uses the max operation for each channel of the feature map.

(4) Fuzzification layer

In this layer, fuzzy inferences are designed to evaluate the degree to which the feature vectors belong to each of the appropriate fuzzy sets. The IF-THEN rule can be expressed as

$Rule[j]$: IF $y_{fusion}[1]$ is $A[1j]$ and $y_{fusion}[2]$ is $A[2]$ and … and y_{fusion} is $A[nj]$,
THEN $y[j] = w_{output}[j]$,

where y_{fusion} is the fused feature vector, $A[ij]$ represents the fuzzy set, $w_{output}[j]$ indicates the weight of the output, and j = 1, 2, 3, …, r is the number of rules. In this study, the Gaussian membership function is adopted. The Gaussian membership function is defined as

$$M[ij] = \exp\left(-\left(y_{fusion}[i] - m[ij]\right)^2 / sd[ij]^2\right), \tag{4}$$

where $M[ij]$ is the Gaussian membership function, and $m[ij]$ and $sd[ij]$ are the mean and standard deviation of the Gaussian membership function, respectively.

(5) Rule layer

By combining the memberships obtained in the previous stage and multiplying them, the firing strengths of the fuzzy rules can be obtained.

$$R\left[j\right]=\prod_{i=1}^{n}M\left[ij\right] \tag{5}$$

(6) Output layer

The output y_{output} is calculated as

$$y_{output}=\sum_{j=1}^{r}R\left[j\right]w_{output}\left[j\right]. \tag{6}$$

2.2 Taguchi method

To design a prediction model with favorable stability and high accuracy, the experience of experts or experimental methods such as trial-and-error, one-factor, and full-factorial experiment are typically used. However, these methods require numerous experiments, which is not only time-consuming but also expensive. By contrast, the Taguchi method is used to rapidly and inexpensively determine an optimum combination of parameters through statistical methods. The Taguchi method has the following six steps: (1) defining the problem, (2) determining control factors and levels, (3) designing an orthogonal table, (4) conducting experiments, (5) analyzing experimental results, and (6) validating experimental results. The flowchart of the Taguchi method is presented in Fig. 3.

In this experiment, we optimized the architectural parameters of the 1D-TCNFN by using the Taguchi method. Eight control factors (the convolution kernel sizes and filter numbers of the three layers, fusion method, and the number of fuzzy rules) were selected to optimize the network architecture. Except for the number of fuzzy rules for which two levels were selected, four levels were selected for the remaining control factors, as presented in Table 1.

As presented in Table 1, a total of eight control factors were identified; of these, seven factors have a level of 4 and the remaining one has a level of 2. If the full-factorial experiment method were to be used to complete this experiment, a total of 32768 ($2^1 \times 4^7$) experiments would be required. Through the design and selection of the orthogonal table, the desired experimental results can be obtained in a small number of experiments. We used the L32 orthogonal table (Table 2) to design this experiment.

To increase the reliability of the experiment, we performed four experiments with each combination, yielding a total of 128 (32 × 4) experiments. The obtained experimental data indicate the effect of each control factor on the experimental results through the signal-to-noise (SN) ratio. Our aim was to effectively improve the accuracy performance by maximizing the SN ratio. The SN ratio is defined as

$$\eta=10\log\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{y_i}\right)^2\right], \tag{7}$$

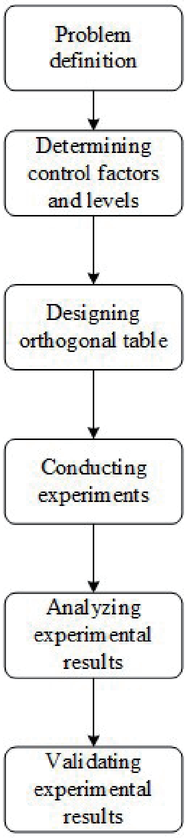

Problem definition
Determining control factors and levels
Designing orthogonal table
Conducting experiments
Analyzing experimental results
Validating experimental results

Fig. 3. Flowchart of Taguchi method.

Table 1
Control factors and their levels.

No.	Control factors	Level 1	Level 2	Level 3	Level 4
A	Convolution Layer 1 Filter	16	32	64	128
B	Convolution Layer 1 Kernel Size	15	30	60	120
C	Convolution Layer 2 Filter	32	64	128	256
D	Convolution Layer 2 Kernel Size	7	15	30	60
E	Convolution Layer 3 Filter	64	128	256	512
F	Convolution Layer 3 Kernel Size	7	15	30	60
G	Fusion Method	CAP	GAP	CMP	GMP
H	Fuzzy Rule	32	64	–	–

CAP: channel average pooling; GAP: global average pooling; CMP: channel max pooling; GMP: global max pooling.

Table 2
Orthogonal array select table.

Orthogonal array	2 levels	4 levels
L8	1–4	1
L16	2–12	1
L16	1–9	2
L16	1–6	3
L16	1–3	4
L32	1	2–9

where η is the S/N ratio (larger is better), n represents the number of experiments, and y indicates the accuracy of the experiment. The optimal parameter combination can be found by analyzing the S/N ratio of each control factor.

3. Experimental Results

In this study, the MIT-BIH Arrhythmia Database was adopted to evaluate the performance of the proposed 1D-TCNFN. This section is divided into three subsections: (1) Introduction to the MIT-BIH Arrhythmia Database, (2) 1D-TCNFN optimized by Taguchi method, and (3) 1D-TCNFN results for arrhythmia detection.

3.1 MIT-BIH Arrhythmia Database

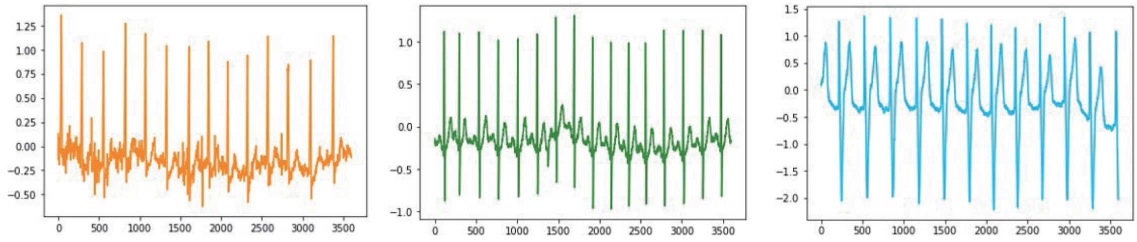
The MIT-BIH Arrhythmia Database contains 48 half-hour dual-channel ECG signals with a sampling frequency of 360 Hz. The signals were obtained from 47 individuals: 25 men aged 32–89 years and 22 women aged 23–89 years. Each record was split into several segments (10 s each). The dataset has a total of 1000 data segments and was divided into training data, verification data, and test data at a ratio of 70:15:15. The detailed detection categories and data segmentation status are presented in Table 3.

A total of 17 different heart rhythm categories are included in these data. Figure 4 presents three heart rhythm signal samples. Figure 4(a) is a normal heart rhythm signal, Fig. 4(b) is an atrial premature beat signal, and Fig. 4(c) is a pacemaker rhythm signal.

Table 3
ECG signal dataset.

No.	Categories	Total	Train	Val	Test
1	Normal sinus rhythm	283	200	47	36
2	Atrial premature beat	66	44	10	12
3	Atrial flutter	20	13	3	4
4	Atrial fibrillation	135	96	21	18
5	Supraventricular tachyarrhythmia	13	9	2	2
6	Pre-excitation (WPW)	21	15	4	2
7	Premature ventricular contraction	133	98	19	16
8	Ventricular bigeminy	55	38	8	9
9	Ventricular trigeminy	13	10	2	1
10	Ventricular tachycardia	10	7	1	2
11	Idioventricular rhythm	10	7	2	1
12	Ventricular flutter	10	6	1	3
13	Fusion of ventricular and normal beat	11	7	3	1
14	Left bundle branch block beat	103	73	11	19
15	Right bundle branch block beat	62	45	8	9
16	Second-degree heart block	10	6	3	1
17	Pacemaker rhythm	45	26	4	14
	Total	1000	700	150	150

Test: testing; Train: training; Val: validation.



(a) (b) (c)

Fig. 4. (Color online) Sample signals of different classes: (a) normal sinus rhythm, (b) atrial premature beat, and (c) pacemaker rhythm.

3.2 Optimal parameters based on Taguchi method

The proposed 1D-TCNFN comprises three convolutional layers, one feature fusion layer, one fuzzification layer, one rule layer, and one output layer, as presented in Table 4. The network architecture contains numerous adjustable hyperparameters, and through the Taguchi experiment, the optimal parameter combination can be identified from these hyperparameters. In this experiment, the L32 ($2^1 \times 4^7$) orthogonal table is used, as shown in Table 5. The orthogonal table can reduce the number of experiments to only 32. To improve stability in the Taguchi experiment, four experiments were performed for each parameter combination. During the Taguchi experiment, the optimizer was Adam, the learning rate was set to 0.001, the batch training size was 128, and the number of iterations for training was 100.

After the Taguchi experiment, the S/N ratio for all control factors was calculated to analyze each factor's contribution. Table 6 shows the six control factors, namely, convolution 1 filter (A), convolution 1 kernel size (B), convolution 2 filter (C), convolution 2 kernel size (D), convolution 3 filter (E), convolution 3 kernel size (F), fusion method (G), and fuzzy rule (H). The top three factors affecting the accuracy of the network were the fusion method (G), convolution 3 kernel size (F), and convolution 2 kernel size (D). By contrast, different fusion methods considerably affected the accuracy of the 1D-TCNFN (Fig. 5). Specifically, the 1D-TCNFN using the global max pooling fusion method outperformed the other fusion methods in terms of arrhythmia detection accuracy.

Table 4
Architecture of 1D-CNFN.

Layer	Layer name	Kernel × Filter	Other layer parameters
1	Input Layer	—	Input_size = 3600
2	Convolution Layer 1	30 × 32	ReLU, Strides = 2
3	Convolution Layer 2	15 × 64	ReLU, Strides = 2
4	Convolution Layer 3	15 × 128	ReLU, Strides = 2
5	Global Average Pooling Layer	—	—
6	Fuzzy Rule Layer	—	Rule = 64
7	Output Layer	1 × 17	Softmax

Table 5
L32 ($2^1 \times 4^7$) orthogonal array and results.

No.	A	B	C	D	E	F	G	H	Y0	Y1	Y2	Y3
1	16	15	32	7	64	7	CAP	32	0.30	0.26	0.24	0.32
2	16	30	64	15	128	15	GAP	32	0.82	0.87	0.89	0.82
3	16	60	128	30	256	30	CMP	32	0.24	0.52	0.24	0.24
4	16	120	256	60	512	60	GMP	32	0.24	0.24	0.24	0.24
5	32	15	32	15	128	30	CMP	32	0.24	0.24	0.24	0.24
6	32	30	64	7	64	60	GMP	32	0.93	0.91	0.91	0.92
7	32	60	128	60	512	7	CAP	32	0.34	0.47	0.44	0.42
8	32	120	256	30	256	15	GAP	32	0.90	0.92	0.93	0.91
9	64	15	64	30	512	7	GAP	32	0.86	0.88	0.89	0.91
10	64	30	32	60	256	15	CAP	32	0.36	0.34	0.45	0.33
11	64	60	256	7	128	30	GMP	32	0.93	0.93	0.97	0.96
12	64	120	128	15	64	60	CMP	32	0.24	0.24	0.24	0.24
13	128	15	64	60	256	30	GMP	32	0.92	0.88	0.92	0.92
14	128	30	32	30	512	60	CMP	32	0.24	0.24	0.24	0.24
15	128	60	256	15	64	7	GAP	32	0.91	0.89	0.82	0.89
16	128	120	128	7	128	15	CAP	32	0.37	0.24	0.28	0.36
17	16	15	256	7	512	15	CMP	64	0.24	0.24	0.24	0.24
18	16	30	128	15	256	7	GMP	64	0.86	0.87	0.92	0.88
19	16	60	64	30	128	60	CAP	64	0.41	0.49	0.52	0.36
20	16	120	32	60	64	30	GAP	64	0.92	0.90	0.89	0.93
21	32	15	256	15	256	60	CAP	64	0.33	0.48	0.42	0.46
22	32	30	128	7	512	30	GAP	64	0.90	0.93	0.93	0.93
23	32	60	64	60	64	15	CMP	64	0.53	0.24	0.51	0.24
24	32	120	32	30	128	7	GMP	64	0.93	0.91	0.91	0.95
25	64	15	128	30	64	15	GMP	64	0.94	0.91	0.90	0.90
26	64	30	256	60	128	7	CMP	64	0.24	0.24	0.24	0.24
27	64	60	32	7	256	60	GAP	64	0.93	0.92	0.95	0.93
28	64	120	64	15	512	30	CAP	64	0.26	0.45	0.39	0.35
29	128	15	128	60	128	60	GAP	64	0.24	0.24	0.24	0.24
30	128	30	256	30	64	30	CAP	64	0.24	0.24	0.24	0.24
31	128	60	32	15	512	15	GMP	64	0.94	0.95	0.91	0.95
32	128	120	64	7	256	7	CMP	64	0.46	0.24	0.46	0.24

CAP: channel average pooling; GAP: global average pooling; CMP: channel max pooling; GMP: global max pooling.

Table 6
S/N ratios of control factors.

	Control factors							
Level	A	B	C	D	E	F	G	H
1	−7.301	−7.398	−5.965	−5.895	−6.251	−5.757	−9.404	−6.649
2	−5.164	−6.299	−5.198	−5.798	−7.236	−5.717	−2.382	−6.286
3	−5.776	−4.975	−7.179	−5.907	−5.255	−6.084	−11.870	–
4	−7.628	−7.198	−7.527	−8.269	−7.127	−8.311	−2.214	–
Delta	2.464	2.423	2.328	2.472	1.981	2.594	9.656	0.3630
Rank	4	5	6	3	7	2	1	8
Best level	2	3	2	2	3	2	4	2
Optimal parameter	32	60	64	15	256	15	GMP	64

CAP: channel average pooling; GAP: global average pooling; CMP: channel max pooling; GMP: global max pooling.

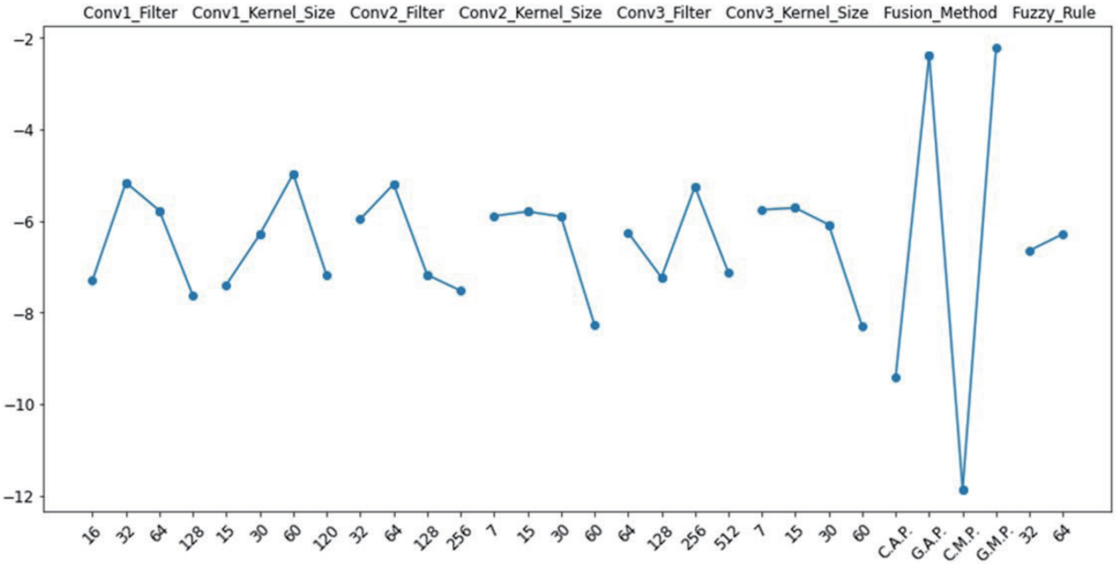


Fig. 5. (Color online) Renderings of control factor.

3.3 Model evaluation for arrhythmia detection

To evaluate the effectiveness of the proposed 1D-TCNFN, four evaluation metrics, namely, accuracy, recall, precision, and F1-score, were used. The definition of each evaluation indicator is as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$Recall = \frac{TP}{TP + FP} \tag{9}$$

$$Precision = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

Here, *TP*, *FP*, *TN*, and *FN* denote true positive, false positive, true negative, and false negative, respectively. The confusion matrix of the detection accuracy of the 1D-TCNFN is presented in Fig. 6 and reveals that the pacemaker rhythm (category 17) is misclassified as the normal sinus rhythm (category 1) for 25% of instances. That is, the characteristics of the pacemaker rhythm signal are similar to those of the normal sinus rhythm signal; thus, the detection accuracy of the 1D-TCNFN for this category is only 66.67%. However, in other arrhythmia categories, the 1D-TCNFN can effectively detect different arrhythmia conditions, with an accuracy ranging from 80 to 100%.

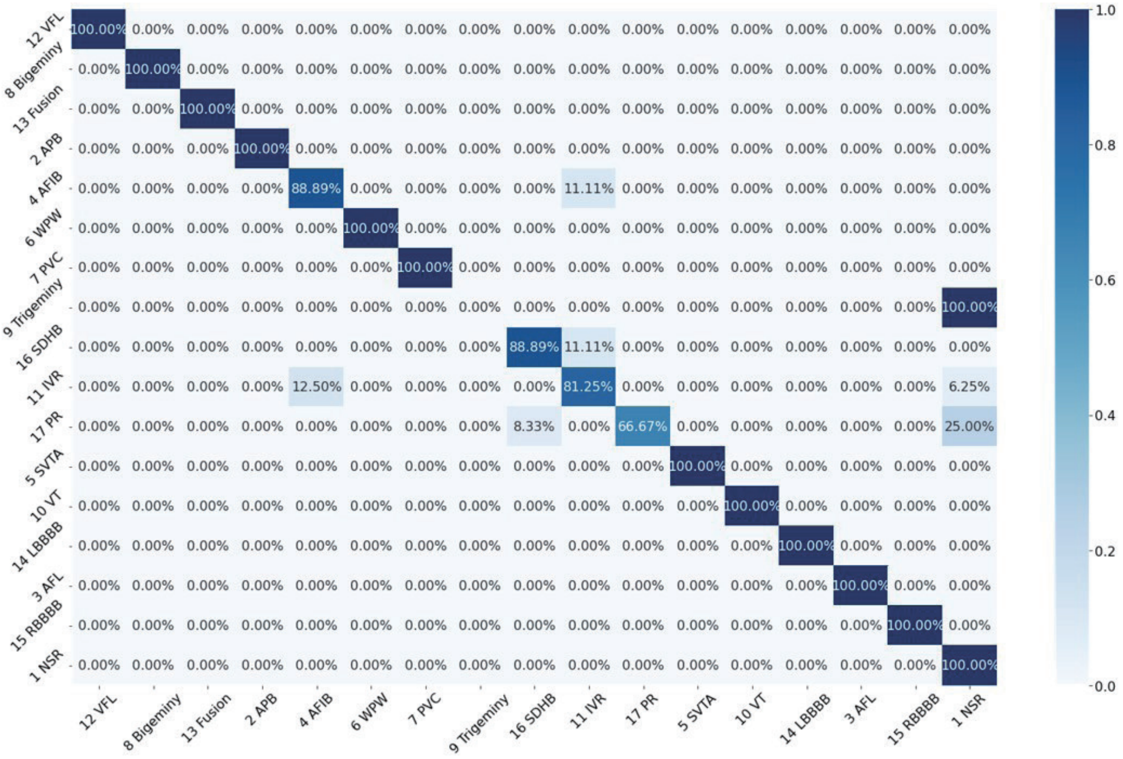


Fig. 6. (Color online) Confusion matrix of 1D-TCNFN for arrhythmia detection.

To demonstrate the superiority of the proposed method, seven deep learning models with different architectures, namely, 1D-LeNet,(23) 1D-AlexNet,(24) 1D-VGG16,(25) 1D-GoogLeNet,(26) 1D-ResNet18,(27) 1D-ResNet50,(28) and 1D-CNFN (without Taguchi optimization), were used to compare the detection performance characteristics (Table 7). When the global average pooling or global max pooling fusion method was employed in the proposed 1D-CNFN, an accuracy of 88.95 or 91.45%, respectively, was achieved. Moreover, the accuracy obtained using the Taguchi

Table 7
Comparison of detection results with different methods.

Method	Feature fusion	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
1D-LeNet	—	25.73	27.48	23.60	25.38
1D-AlexNet	—	42.13	42.74	41.20	41.96
1D-VGG16	—	47.20	53.44	42.67	44.47
1D-GoogLeNet	—	77.87	80.93	75.33	78.03
1D-ResNet18	—	84.00	86.02	83.07	84.50
1D-ResNet50	—	48.40	51.37	47.07	49.01
1D-CNFN	CAP	37.63	41.37	28.53	33.72
	GAP	88.95	94.30	84.13	88.49
	CMP	39.60	48.57	28.53	35.91
	GMP	91.45	94.10	91.20	92.62
1D-TCNFN	GMP	93.95	96.03	93.47	94.30

CAP: channel average pooling; GAP: global average pooling; CMP: channel max pooling; GMP: global max pooling.

method was 2.5 percentage points higher than that of the original 1D-CNFN. Thus, the results indicated that the proposed 1D-TCNFN outperforms other state-of-the-art deep learning models in arrhythmia detection.

4. Conclusions

We propose a 1D-TCNFN for the detection of 17 types of arrhythmia. The method can detect abnormal ECG signals by using a long-term signal segment, and its application can potentially reduce the workload of medical staff. The proposed 1D-TCNFN combines convolution layers and a neuro-fuzzy network; this network not only automatically extracts features from raw data but also reduces the excessive number of parameters resulting from using fully connected layers. We used the Taguchi method to optimize the architecture and performance of the 1D-TCNFN. The open MIT-BIH Arrhythmia Database was subsequently adopted to evaluate the performance of the proposed architecture. The performance of the proposed 1D-TCNFN is superior to various state-of-the-art deep learning networks, yielding the accuracy and F1-score of 93.95 and 94.30%, respectively. In future research, we intend to port the proposed method to embedded systems for edge computing. In addition, we intend to introduce an attention mechanism to improve the overall model detection performance for difficult-to-identify ECG signals.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan for financially supporting this research under Contract No. MOST 110-2221-E-167-031-MY2.

References

1 M. S. Thaler: The Only Ekg Book You'll Ever Need (Lippincott Williams & Wilkins, Philadelphia, 2007) 10th ed.
2 D. Mozaffarian, E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. R. Das, S. d. Ferranti, J. P. Després, H. J. Fullerton, V. J. Howard, M. D. Huffman, C. R. Isasi, M. C. Jiménez, S. E. Judd, B. M. Kissela, J. H. Lichtman, L. D. Lisabeth, S. Liu, R. H. Mackey, D. J. Magid, D. K. McGuire, E. R. MohlerIII, C. S. Moy, P. Muntner, M. E. Mussolino, K. Nasir, R. W. Neumar, G. Nichol, L. Palaniappan, D. K. Pandey, M. J. Reeves, C. J. Rodriguez, W. Rosamond, P. D. Sorlie, J. Stein, A. Towfighi, T. N. Turan, S. S. Virani, D. Woo, R. W. Yeh and M. B. Turner: Circulation **133** (2016) 38. https://doi.org/doi:10.1161/CIR.0000000000000350
3 T. Barill and S. L. Inc: The Six Second Ecg: A Practical Guide to Basic and 12 Lead Ecg Interpretation (SkillStat Learning Incorporated, North Vancouver, 2012).
4 B. Hedén, M. Ohlsson, L. Edenbrandt, R. Rittner, O. Pahlm, and C. Peterson: Am. J. Card. **75** (1995) 929. https://doi.org/10.1016/s0002-9149(99)80689-4
5 Ö. Yıldırım, P. Pławiak, R.-S. Tan, and U. R. Acharya: Comput. Biol. Med. **102** (2018) 411. https://doi.org/10.1016/j.compbiomed.2018.09.009
6 P. Pławiak: Expert Syst. Appl. **92** (2018) 334. https://doi.org/https://doi.org/10.1016/j.eswa.2017.09.022
7 A. S. Alvarado, C. Lakshminarayan, and J. C. Principe: IEEE Trans. Biomed. Eng. **59** (2012) 1641. https://doi.org/10.1109/TBME.2012.2191407
8 J. Mateo, A. M. Torres, A. Aparicio, and J. L. Santos: Comput. Electr. Eng. **53** (2016) 219. https://doi.org/10.1016/j.compeleceng.2015.12.015
9 Z. Li and H. Zhang: Front. Cardiovasc. Med. **8** (2021) 616585. https://doi.org/10.3389/fcvm.2021.616585

10 K. Padmavathi and K. S. Ramakrishna: Procedia Comput. Sci. **46** (2015) 53. https://doi.org/10.1016/j.procs.2015.01.053

11 S. M. Usman, M. Usman, and S. Fong: Comput. Math. Methods Med. **2017** (2017) 9074759. https://doi.org/10.1155/2017/9074759

12 G. Thambiraj, U. Gandhi, U. Mangalanathan, V. J. M. Jose, and M. Anand: Biomed. Signal Process. Control **60** (2020) 101942. https://doi.org/10.1016/j.bspc.2020.101942

13 J.-S. Wang, W.-C. Chiang, Y.-L. Hsu, and Y.-T. C. Yang: Neurocomputing **116** (2013) 38. https://doi.org/10.1016/j.neucom.2011.10.045

14 J. H. Abawajy, A. V. Kelarev, and M. Chowdhury: Comput. Methods Programs Biomed. **112** (2013) 720. https://doi.org/10.1016/j.cmpb.2013.08.002

15 J. Schmidhuber: Neural Networks **61** (2015) 85. https://doi.org/10.1016/j.neunet.2014.09.003

16 C. Goller and A. Kuchler: Proc. Int. Jt. Conf. Neural Netw. (ICNN'96) **1** (1996) 347. https://doi.org/10.1109/ICNN.1996.548916

17 Y. LeCun, Y. Bengio, and G. Hinton: Nature **521** (2015) 436. https://doi.org/10.1038/nature14539

18 J. S. R. Jang: IEEE Trans. Syst. Man Cybern.: Syst. **23** (1993) 665. https://doi.org/10.1109/21.256541

19 W. Limei, Z. Zongxue, and L. Xiaoying: 2017 29th Chinese Control and Decision Conf. (CCDC) (2017) 4716. https://doi.org/10.1109/CCDC.2017.7979330

20 T. T. Khuat and B. Gabrys: Neurocomputing **386** (2020) 110. https://doi.org/10.1016/j.neucom.2019.12.090

21 D. S. Karna and R. Sahai: Int. J. Eng. Math. **1** (2012) 1.

22 G. B. Moody and R. G. Mark: IEEE Eng. Med. Biol. Mag. **20** (2001) 45. https://doi.org/10.1109/51.932724

23 L. Wan, Y. Chen, H. Li, and C. Li: Sensors **20** (2020) 1693. https://doi.org/10.3390/s20061693

24 Q. Yang, L. Zou, K. Wei, and G. Liu: Comput. Biol. Med. **140** (2022) 105124. https://doi.org/10.1016/j.compbiomed.2021.105124

25 M. Sallem, A. Ghrissi, A. Saadaoui, and V. Zarzoso: Comput. Cardiol. (2020) 1. https://doi.org/10.22489/CinC.2020.339

26 C. Min, G. Wen, Z. Yang, X. Li, and B. Li: Energies **12** (2019). https://doi.org/10.3390/en12152882

27 Y. Qiao, X. Li, J. Wang, S. Ji, T. Hirtz, H. Tian, J. Jian, T. Cui, Y. Dong, X. Xu, F. Wang, H. Wang, J. Zhou, Y. Yang, T. Someya, and T.-L. Re: Small **18** (2022) 2104810. https://doi.org/10.1002/smll.202104810

28 T. Jian, Y. Gong, Z. Zhan, R. Shi, N. Soltani, Z. Wang, J. G. Dy, K. R. Chowdhury, Y. Wang, and S. Ioannidis: IEEE Trans. Mob. Comput. (2021) 1. https://doi.org/10.1109/TMC.2021.3064466

About the Authors

Jiarong Li received her B.S. degree in electrical engineering from Huaihai Institute of Technology in 1993 and her M.S. degree in power electronics and power transmission from Nanjing University of Aeronautics and Astronautics in 2001. Currently, she is an associate professor of the Electrical Engineering College, Yancheng Institute of Technology. Her current research interests are in power systems and new energy systems. (lijr@ycit.cn)

Jyun-Yu Jhang received his B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2015 and his Ph.D. degree from the Institute of Electrical and Control Engineering in 2021. Currently, he is an assistant professor of the College of Intelligence, National Taichung University of Science and Technology, Taichung, Taiwan. His current research interests include fuzzy logic theory, type-2 neural fuzzy systems, evolutionary computation, machine learning, computer vision, and applications. (jyjhang@nutc.edu.tw)

Cheng-Jian Lin received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986 and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. Currently, he is a chair professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C., and the dean of Intelligence College, National Taichung University of Science and Technology, Taichung, Taiwan, R.O.C. His current research interests are in machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots. (cjlin@ncut.edu.tw)

Xue-Qian Lin is a graduate student in the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C. His current research interests are in fuzzy neural network, image processing, and machine learning. (th0rnlin1412@gmail.com)