A Deep Learning Approach for Patient-Specific ECG Signal Classification with 1-D CNNs

Saima Ali[*], Irfan Ahmed[†], Abid Iqbal[‡], Salman Ilahi Siddiqui[§], Amaad Khalil[**]

Abstract

Cardiovascular diseases (CVDs) still symbolize the leading cause of death in the world. Electrocardiogram (ECG) is a primary diagnostic modality for diagnosing conditions, but its interpretation requires clinical expertise and is prone to inter-observer variability. Recent advances in deep learning have enabled the development of automated ECG classification algorithms and, thus, have offered more stable and effective decision-support mechanisms. This paper presents a small one-dimensional Convolutional Neural Network (1D-CNN) used to detect four types of rhythm in patient-specific ECG signals: Normal, Atrial Fibrillation, Other, and Noise. The network uses very little preprocessing of the raw ECG recordings, making it an end-to-end learning pipeline. Using the PhysioNet/Computing in Cardiology Challenge 2017 dataset, three CNN configurations were evaluated. The proposed framework achieved an overall accuracy of 85.4%, a macro-F1 score of 0.81, and a weighted F1 Score of 0.803. Results demonstrate that a lightweight CNN can achieve competitive performance compared to more complex state-of-the-art methods, while maintaining simplicity, reproducibility, and potential for clinical deployment.

Keywords*:* Electrocardiogram *(*ECG), Convolutional Neural Network (CNN), 1D Convolution, ECG Signals.

Introduction

Due to the rapid advancement of innovation and the expanded use of compact sensing devices, a significant amount of biomedical data is recorded daily to monitor and assess the physiological state of the human body (Ding et al., 2025) (Ahmed et al., 2023). These biomedical signs measure the physiological functions of various organs, such as the heart, brain, muscles, cornea, and so on (Wagner & Strauss, 2014). They are generally acquired by setting at least one anode on the organ of intrigue. Electrocardiogram (ECG) and Electroencephalogram (EEG) are the most

[*]Department of Electrical Engineering, University of Engineering & Technology, Jalozai Campus, Peshawar 24100, Pakistan, saimaaaali98@gmail.com

[†]Department of Electrical Engineering, University of Engineering & Technology, Jalozai Campus, Peshawar 24100, Pakistan, irfanahmed@uetpeshawar.edu.pk

[‡]Corresponding Author: Department of Computer Engineering, College of Computer Sciences and Information Technology, King Faisal University, AlAhsa 31982, Saudi Arabia, aaiqbal@kfu.edu.sa

[§]Department of Electrical Engineering, University of Engineering & Technology, Peshawar 25120, Pakistan, salman.ilahi@uetpeshawar.edu.pk

[**]Department of Computer Systems Engineering, University of Engineering & Technology, Peshawar 25120, Pakistan, amaadkhalil@uetpeshawar.edu.pk

well-known biomedical signals recorded from the heart and brain, respectively (Ding et al., 2024).

The electrical activity of one's heart is recorded using an ECG. It is a common technique used to assess heart status and identify heart issues in various situations (Samant, 2023). ECG is a diagnostic test that measures the electrical activity of the heart, and a specialist can diagnose heart irregularities by interpreting it (Bijl et al., 2022). For an individual, the ECG signal has a standard waveform, and any adjustment in heart rate is reflected in it (Perloff & Marelli, 2012).

The expert looks at characteristic features of the ECG waveform when classifying heartbeats. These include R-R interval, P-wave, QRS complex and T-wave (Asl et al., 2008). Various components that constitute a normal sinus rhythm ECG waveform are shown in Figure 1.

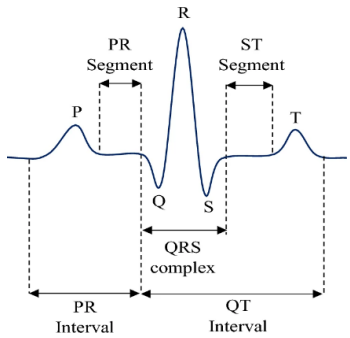


Figure 1: Components of an ECG Signal.

Cardiovascular Disease (CVD) is a constant medical condition that is characterized by risk and intense onset. It is a genuine danger to an individual's wellbeing. As of now, there is an excess of 300 million people with CVD in China. The mortality of CVD in provincial territories and cities is 44.8% and 41.9%, respectively. It positions as a matter of first importance incessant maladies and is sufficient to seek our attention (Ma et al., 2020).

ECG signals have the benefits of being basic, helpful, safe, and non-hazardous to patients. ECG is commonly used to analyze CVDs. Notwithstanding, it is difficult for specialists to properly examine the data due to the sheer volume and complexity of ECG data (Jin, 2018). Besides, the flimsiness of ECG waveforms is not just reflected in the changeability of their shape, but is also closely associated with time, individual, condition, and other factors. In this way, it represents a key advance in precisely separating the salient features of ECG waveforms (Joshi et al., 2009).

The objective of this research is to process ECG for the classification of heart illness. For this reason, we use supervised deep learning procedures and select the CNN, which is the most suitable for feature extraction and classification. Before the model training and test phases, a dataset of ECG signals is passed through a preprocessing block to extract key features and remove redundant information. This study focuses on the classification of ECGs into four zones: typical sinus beat, arrhythmic, another rhythm, and extremely noisy. The industrial significance of the proposed work lies in therapeutic hardware engineering, where efficient devices can be designed to provide handheld health monitoring and diagnostics systems.

Literature Review

ECG signal classification has undergone significant changes over the last 20 years, moving away from traditional signal-processing methods toward high-quality deep learning models. The development reflects improvements in computing power and the increasing accessibility of large-scale healthcare data. The inherent problem of ECG classification lies in the fact that it is hard to clearly identify the cardiac arrhythmia and classify it, and that it is challenging to consider that there is much inter-patient variability in the morphology of signals, the characteristics of their amplitude, and the physiological background (Mahajan et al., 2024). Early methods of ECG classification were very much dependent on manual feature extraction methods that were coupled with standard machine-learning classifiers. These were multi-stage processing pipelines that usually included noise reduction/filtering, QRS complex detection, extraction of morphological features, and statistical analysis.

Recent advances have explored various architectural innovations aimed at improving classification accuracy and computational speed simultaneously. A more recent example of such models is hybrid CNN-BiLSTM networks, which combine convolutional feature extraction and recurrent temporal modeling, achieving final results of 98% accuracy and 91% sensitivity and specificity in five-class arrhythmia classification tasks (Kalatehjari et al., 2025). These architectures leverage both the local pattern-detection capabilities of convolutional layers and the temporal dependency modelling capabilities of recurrent layers. Recent methods of one-dimensional CNN (1D-CNN) classification in ECG have aimed to address problems such as vanishing gradients in deep networks, the need for attention mechanisms, and computational efficiency optimization (Bammara and Mousselmal, 2025). Apart from the recently used heuristic based methods used for many 1D signals (Ahmed et al., 2022), deep residual neural network methods add skip connections to allow deeper

models to be trained with gradient flow, and achieve improved classification performance on a number of arrhythmia types. The combination of attention mechanisms with traditional CNN architectures is a major advancement in the representation of both localized time variations and global-scale dependencies in ECG signals. Transformer-based designs use self-attention to better capture long-term dependencies than traditional CNNs alone, thereby overcoming shortcomings in capturing inter-heartbeat relationships with significant diagnostic potential (Ikram et al., 2025). The presence of these hybrid CNN-Transformer models demonstrates how modern architectures can leverage the complementary capabilities of different deep-learning paradigms. Self-Operational

Neural Networks represent a significant novelty and a continuation of standard CNN designs, incorporating activation functions that adapt to the specifics of ECG signals and are learnable (Zahid et al., 2022). The initial experiments show that 1D Self-ONNs outperform traditional 1D-CNNs at the same computational complexity, a promising sign for future architectural design. Although conventional approaches have shown decent performance, their reliance on expert feature engineering and limited generalization have significantly limited their clinical applicability. Deep-learning methods, especially patient-specific CNN models, are effective in overcoming these limitations by learning global and individual-specific signal properties. However, even with these advancements, including hybrid CNN-BiLSTM and CNN-Transformer models, there are still considerable problems with generalization across heterogeneous populations, computational complexity, and a lack of annotated patient-specific information. These limitations highlight the need for not only accurate models but also lightweight, flexible, and clinically scalable models. This paper is a natural extension of this trend, as it proposes a new 1D CNN framework that combines efficiency, adaptability, and robustness.

The novelty of this work lies in the design and validation of a compact 1D–CNN architecture that combines efficiency with competitive classification performance. Unlike many recent hybrids or deeply stacked architectures, the proposed model demonstrates that a streamlined convolutional pipeline can achieve reliable results without excessive computational burden. The framework integrates end-to-end learning, minimal preprocessing, and reproducible reporting protocols, which together provide transparency and ease of replication. Moreover, the study offers a systematic evaluation across three CNN setups, highlighting the stability of the approach under different architectural choices. This

strengthens its contribution as both a methodological and practical advance.

Methodology

ECG Preprocessing

ECG preprocessing is an essential step for signal analysis as it provides considerable information on physiological processes in the human body, especially regarding cardiac activity. In our work, we have employed several preprocessing techniques to increase precision, including feature engineering, feature selection, data transformation, and preprocessing. In our study, we examined the principles of ECG signal processing, including preprocessing, ECG database, feature extraction, and classification using a variety of statistical analysis techniques.

As ECG recordings are prone to various artefacts (additive white noise and low-frequency baseline wander) preprocessing is necessary to reduce such perturbations which otherwise may corrupt the features used later in classification. Here, the raw data, which is always incomplete, is processed into a form the algorithm can digest; without this processing, the raw data would be processed directly by the model, yielding unreliable estimates. Based on this, proper model training is preconditioned by preprocessing. Feature engineering allows one to extract additional information from available observations. The creation of new descriptors can reveal additional explanatory power for the variations in the training data, thereby improving prediction. To determine an ideal set of variables that fully describe the relationship between the predictors and the response, feature selection methods are used. The methods used by the researchers to extract ECG features include the Discrete Wavelet Transform (DWT), Continuous Wavelet Transform (CWT), Discrete Cosine Transform (DCT), Short-Time Fourier Transform (STFT), Discrete Fourier Transform (DFT), the Pan-Tompkins algorithm, and Independent Component Analysis (ICA). Unity Standard Deviation (SD) scaling and z-score normalization are standard procedures for feature standardization (Taha et al., 2016).

DWT is beneficial for preprocessing and feature extraction in signal analysis, and it provides a means to decompose ECG signals into multiple temporal scales. While effective at precisely isolating QRS complexes from other ECG segments, its practical utility for detecting low-frequency waves, such as P and T waves, is limited. The DWT is defined by:

$$T_{m,n} = \int_{-\infty}^{+\infty} \psi_{m,n}(t)dt \quad (1)$$

where, m and n are signal scale and location, respectively.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are hierarchical feed-forward systems inspired by the biological visual cortex. Unlike traditional neural networks that rely on fully connected layers, CNNs employ learnable filters applied to sub-regions of the input data. Within the network's architecture, the output of each layer is computed using a consistent weight matrix (kernel) and an activation function applied to each data block. CNNs are special types of neural networks that have emerged as the most extensively used deep learning models in recent years. These have made tremendous progress in two-dimensional (2D) image processing. In medicine, CNNs are commonly used in medical imaging for tasks such as tumor detection in brain MRIs. The use of CNNs, however, has had a relatively low adoption rate in bio-signal processing, especially for one-dimensional physiological signals such as electrocardiography (ECG).

Each convolutional layer contains several parameters for its filters. Although these filters cover the entire depth, they are usually small compared to the input volume. In CNNs, the initial argument of the convolution operation is the input, while the subsequent argument is the kernel size. The generated feature map serves as the input for the next layer. In the field of Machine Learning (ML), data is commonly represented as a multidimensional array, and the filter's kernel is typically a multidimensional array of parameters learned by the learning algorithm. An activation function (commonly sigmoid or tanh) is applied to introduce non-linearity to this layer. If the kernel size is (k_m,k_n), the two-dimensional input data size is (m, n), and (s_x, s_y) denote the convolution strides, then the output values $d_{x,y}$ are computed using the following equation:

$$d_x = h\left(b + \sum_{i=0}^{k_m-1} \sum_{j=0}^{k_n-1} d_{x_0, y_0} * w_{i,j}\right) \quad (2)$$

where, $x_0 = x.s_x + i$ and $y_0 = y.s_y + j$ are the input horizontal and vertical positions and $w_{i,j}$ is the weight of the kernel at position (i, j).
The size of the output layer, m_0 and n_0, are given by:

$$\begin{aligned} m_0 &= \frac{m-kx}{s_x} + 1 \\ n_0 &= \frac{n-ky}{s_y} + 1 \end{aligned} \quad (3)$$

In the majority of CNN implementations, a pooling layer typically follows the convolutional layers. The principal function of the pooling layer is the reduction of the spatial dimensions of the input, resulting in a decrease in the number of parameters and, consequently, the computational load within CNNs. Each rectangular data block processed by this layer yields a single output. While several methods exist, the most common approach involves selecting the maximum value within the

block. Therefore, the number of features will be reduced by a factor of 4 for a block of size 2×2.

Every input from one layer in a neural network is coupled to every activation unit in the subsequent layer, indicating that the layers are fully connected. In many prevalent ML models, the final layers are typically fully connected, integrating features extracted by preceding layers to produce the ultimate output. This layer is often the second most computationally intensive, following the convolutional layer.

In the SoftMax classifier, the computed probabilities range from 0 to 1, and their sum equals 1. This is mathematically expressed as:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{K=1}^{K} e^{z_K}} \quad (4)$$

Data Description and Preprocessing

For this study, the PhysioNet/Computing in Cardiology Challenge 2017 single–lead ECG training dataset was used. It has four annotations: Normal (N), Atrial Fibrillation (A), Other (O), and Noisy (∼), as shown in Table 1.

Table 1: Arrhythmia classes, annotation codes, and dataset support.

Class	Annotation	Count
Normal rhythm	N	n_{N}
Atrial fibrillation	A	n_{A}
Other rhythm	O	n_{O}
Noisy	∼	$n_{\sim}$
Total		$N = n_{\mathrm{N}} + n_{\mathrm{A}} + n_{\mathrm{O}} + n_{\sim}$

For heartbeat extraction, fixed–length ECG segments were classified without explicit R–peak detection or beat cropping. This avoids error propagation from misdetections and lets the CNN learn morphology directly from raw segments. Records (in the format *Axxxxxx.mat*) were loaded and each waveform was periodically tiled to $T = 10{,}100$ samples. A datas-et–wise z–score $(X - \mu)/\sigma$ was applied, and the inputs were reshaped to $(T, 1)$. Labels were mapped as $\{\mathrm{N} \to 0, \mathrm{A} \to 1, \mathrm{O} \to 2, \sim \to 3\}$. A random permutation yielded a $90\%/10\%$ training/validation split. Table 2, below is the pseudocode that explains the working of the algorithm.

Training for Classifying Patients

A patient–specific framing (segment–level classification per patient) was adopted. The base training protocol used a four–way SoftMax with categorical cross–entropy, Adam, batch size 275, and 20 epochs. Table 3 shows the base training protocols and hyperparameters.

Table 2: Algorithm 1 ECG classification with 1D–CNN.

1: Input: training2017/, REFERENCE.csv, *fixed length* $\boldsymbol{T = 10,100}$
2: *Read labels; map (N,A,O,~)* $\rightarrow \{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{3}\}$*; one–hot encode*
3: for *each Axxxxxx.mat do*
4: $\boldsymbol{s} \leftarrow$ *val[0,:]; if* $\lvert \boldsymbol{s} \rvert \geq \boldsymbol{T}$ *then* $\boldsymbol{x} \leftarrow \boldsymbol{s}[:\boldsymbol{T}]$ *else tile(****s****) and truncate to* $\boldsymbol{T}$ *STATE append* $\boldsymbol{x}$ *to* $\boldsymbol{X}$ *and one–hot label to* $\boldsymbol{Y}$
5: end for
6: *Standardize* $\boldsymbol{X} \leftarrow (\boldsymbol{X} - \boldsymbol{\mu})/\boldsymbol{\sigma}$*; reshape to* $(\boldsymbol{T}, \mathbf{1})$
7: *Shuffle; split into train (*$\mathbf{90}\%$*) and validation (*$\mathbf{10}\%$*)*
8: *Build 1D–CNN (Conv1D(128,55)→Pool(10)→Drop(0.5)→Conv1D(128,25)→Pool(5)→Drop(0.5)→Conv1D(128, 10)→Pool(5)→Drop(0.5)→Conv1D(128,5)→GAP→Dense(256)→Drop(0.5)→Dense(128)→Drop(0.5→ Dense(64)→Drop(0.5)→Dense(4,softmax))*
9: *Train (Adam, CE, batch* $\mathbf{275}$*, epochs* $\mathbf{20}$*)*
10: *Evaluate validation; compute accuracy, precision, recall, F1; save confusion matrix and logs*

Table 3: Base training protocol and hyperparameters.

Item	Setting
Input length	$T = 10{,}100$ (truncate/tile)
Normalization	Global standardization $(X - \mu)/\sigma$
Optimizer / Loss	Adam (default) / Categorical cross–entropy
Batch size / Epochs	275 / 20 (base protocol)
Activations	ReLU (hidden), Softmax (output)
Regularization	Dropout 0.5 after each block and dense layer
Split	Random shuffle; $90\%/10\%$ train/validation
Label map	N→0, A→1, O→2, ~→3 (one–hot)

Experiments

We performed three different experiments using CNN algorithms by changing parameters such as changing number of layers, size of filters etc. To avoid difficulties with overfitting, we used a dropout approach. During training, the input value is arbitrarily set to 0 with a specified probability (i.e., dropping units and their connections). After the convolutional and totally associated layer, dropout layers with a likelihood of 0.5 were implemented through the whole model.
The details of the setup of all three experiments are given as follows and presented in Table 4.

Case I

In the first analysis, the total number of layers was 21. 5 convolutional layers were utilized having 128 neurons. The filter size was 55, 27, 21, 9, and 5 in first, second, third, fourth, and 5th layers, respectively. 5 pooling layers were utilized. Initially, 4 pooling layers were max-pooling and the last pooling layer was Global ave-pooling. The window size was 10,5,5,5 in first, second, third, and fourth pooling layers, respectively. 7 dropout layers were utilized,4 fully connected layers having one output layer was utilized. Number of neurons were 256,128 and 64 respectively in first three fully connected layers and 4 neurons were

in the output layer. The training size was set to 90%, the loss was calculated using categorical cross entropy, the weight was optimized using Adam's optimizer, the batch size was set to 275, and the number of epochs was set to 20.

Table 4: High–level specifications of the three CNN setups.

	Case I	Case II	Case III
Total layers	21	22	21
Conv layers (filters)	5 (128 each)	4 (128 each)	5 (128 each)
Kernel sizes (conv)	55, 27, 21, 9, 5	57, 27, 11, 5	59, 27, 21, 11, 10
Pooling	Max: 10,5,5,5; last GAP	Max: 10,5,5; last GAP	Max: 5,5,5,5,4 (no GAP)
Dropout layers	7	6	7
Batch normalization	–	4 layers	–
Dense widths	256, 128, 64, (4)	256, 128, 64, (4)	256, 128, 64, (4)
Train split	90%	85%	90%
No. of Epochs	20	20	20
Batch size	275	275	275
Loss / Optimizer	CE / Adam	CE / Adam	CE / Adam

Case II

In the second analysis, absolute numbers of layers were 22. 4 convolutional layers were utilized having 128 neurons. The channel size was 57, 27, 11 and 5 in first, second, third and fourth layers individually. 4 pooling layers were utilized. Initially, 3 pooling layers were max-pooling and last pooling layer was Global ave-pooling. Window size was 10, 5, and 5 in first, second and third pooling layers, respectively. 6 dropout layers were utilized. 4 Batch Normalization layers were utilized, 4 fully connected layers having one output layer was utilized. Number of neurons were 256,128 and 64 separately in initial three fully connected layers and 4 neurons were in the output layer. Preparing size was 85%, categorical cross entropy was utilized for calculation loss. Adam's optimization was utilized for weights optimization; batch size was set to 275. Number of epochs were set to 20.

Case III

In the last analysis, the absolute number of layers was 21. 5 convolutional layers were utilized having 128 neurons. The channel size was 59,27,21,11 and 10 in 1st, 2nd, 3rd, 4th, and 5th layers, respectively. 5 pooling layers were utilized. All pooling layers were max pooling. The window size was 5, 5, 5, 5, and 4 in 1st, 2nd, 3rd fourth and 5th pooling layers, respectively. 7 dropout layers were used, 4 fully connected layers having one output layer was utilized. Number of neurons were 256,128 and 64 separately in initial three fully connected layers and 4 neurons were in the output layer. The training size was 90%, categorical cross entropy was utilized for figuring calculation loss. Adam's optimizer was utilized

for weight optimization, the batch size was set to 275, and epochs were set to 20.

Performance Metrics

We reported the overall accuracy together with per–class precision, recall, and F1, and we summarized them as macro and weighted averages for all analyses. For a class i with counts TP_i, FP_i, FN_i and support n_i (total $N = \sum_i n_i$), the per-class precision, recall, and F1 are calculated as:

$$\mathrm{Precision}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \tag{5}$$

$$\mathrm{Recall}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{6}$$

$$\mathrm{F1}_i = \frac{2\,\mathrm{Precision}_i\,\mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i} \tag{7}$$

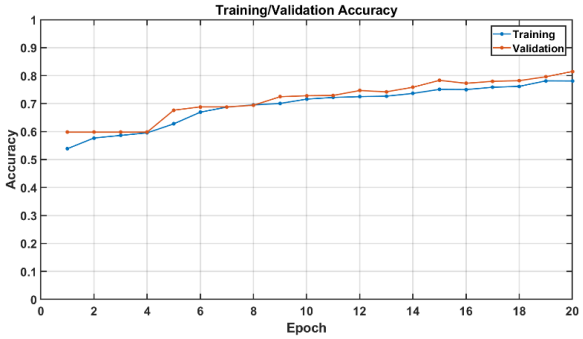
Results and Discussion

We evaluated three 1-D CNN configurations (Case I-III) on the PhysioNet/CinC 2017 four-class task (N, A, O, ~), following the common protocol described in the previous section. Figure 2 and 3 shows the accuracy and loss trajectories, respectively, and Figure 4 shows the confusion matrices. Table 5 summarizes the validation metrics; Tables 6–8 report per-class precision/recall/F1 and supports; and Table 9 reports the challenge scoring proxy (mean F1 over {N, A, O}).

It can be observed that Experiment 1 achieved the strongest validation accuracy (0.8148) and weighed F1 (0.8025), with the highest challenge-proxy F1 over {N,A,O} (0.7555). Experiment 2 delivered a more balanced treatment of the noise class (~) (recall 0.767) but a lower overall accuracy (0.7758) driven by reduced recall for class O (0.447). Experiment 3 showed the weakest macro performance (macro F1 0.5098), dominated by failure to retrieve the noise class (~: precision/recall $= 0/0$ on the validation split). Across all three models, class N is consistently well recalled ($\approx$ 0.96–0.97), while A and O are the principal error modes, reflecting known difficulties in separating atrial fibrillation and heterogeneous "other" rhythms when trained with simple class-balanced cross-entropy and no explicit imbalance handling.

Relative to *Case II* and *Case III*, *Case I's* filter schedule {55,25,10,5} followed by global averaging appears to strike the best locality/context trade-off for this dataset: it preserves high recall for N (0.961) without sacrificing AF detection (A recall 0.831) and avoids over-suppressing O (O recall 0.581). Experiment 2's stronger performance on the noise class (recall 0.767) coincides with a modest drop in O, plausibly reflecting different inductive bias from its configuration (as specified in

the experimental design) that encourages aggressive denoising at the cost of borderline O segments. *Case III*, which pools uniformly and compresses aggressively, under-recognizes minority patterns, consistent with its low macro scores and the ~ miss.

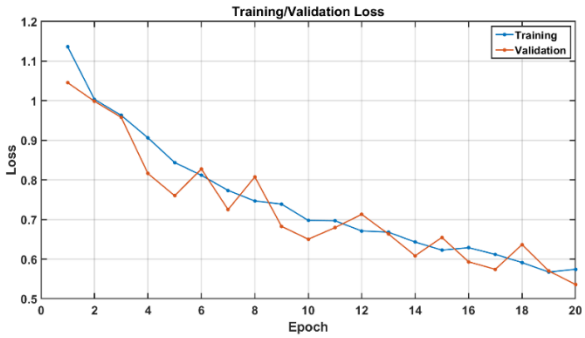


(a)

(b)

(c)

Figure 2: Comparison of the accuracy curves across the three configurations.



(a)

(b)

(c)

Figure 3: Comparison of the loss curves across the three configurations.

Table 5: Validation summary metrics across experiments.

Exp.	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	Precision (w)	Recall (w)	F1 (w)
Case I	0.8148	0.7679	0.6557	0.6619	0.8251	0.8148	0.8025
Case II	0.7758	0.7447	0.7009	0.7047	0.7769	0.7758	0.7566
Case III	0.7585	0.5309	0.5203	0.5098	0.7314	0.7585	0.7249

Table 6: Per-class validation metrics for Case I.

Class	Precision	Recall	F1	Support
N	0.8419	0.9608	0.8974	510
A	0.5684	0.8308	0.6750	65
O	0.8614	0.5813	0.6942	246
~	0.8000	0.2500	0.3810	32

Table 7: Per-class validation metrics for Case II.

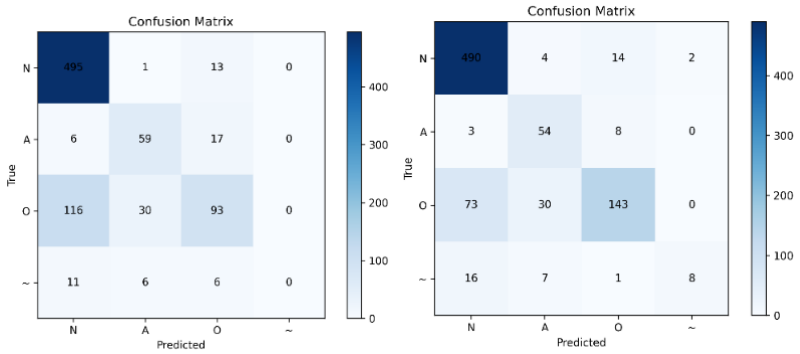
Class	Precision	Recall	F1	Support
N	0.7819	0.9650	0.8639	743
A	0.7959	0.6240	0.6996	125
O	0.7783	0.4472	0.5680	369
~	0.6226	0.7674	0.6875	43

Table 8: Per-class validation metrics for Case III.

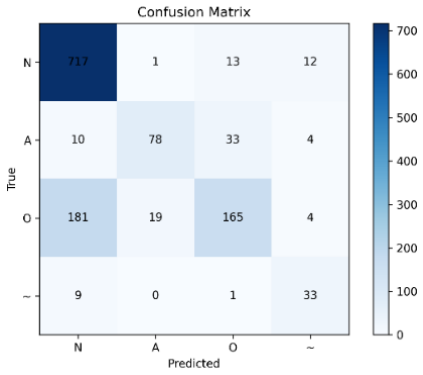
Class	Precision	Recall	F1	Support
N	0.7882	0.9725	0.8707	509
A	0.6146	0.7195	0.6629	82
O	0.7209	0.3891	0.5054	239
~	0.0000	0.0000	0.0000	23

Table 9: Challenge-proxy score: mean F1 over {N, A, O}

Experiment	Mean F1 (N, A, O)
Case I	0.7555
Case II	0.7105
Case III	0.6797



(a) *(b)*



(c)

Figure 4: Confusion matrices for (a) Case I, (b) Case II, and (c) Case III

Comparison with the Contemporary Methods

To position our compact 1-D CNN against peer approaches on the CinC/PhysioNet 2017 arrhythmia task, we rely on the macro–F1 over {N, A, O} and restrict the comparison to *internal-validation* results reported on the public training set (i.e., cross-validation or held-out splits), which is methodologically comparable to our 90/10 held-out protocol. This avoids conflating results from the hidden-test leaderboard with in-sample estimates and provides a fair baseline-oriented perspective. Table 10 summarizes the comparison; we refer the reader to the original papers for implementation details beyond the brief model descriptors.

Table 10: Internal-validation comparison on CinC/PhysioNet 2017 training set

Method	Model/Features	Eval. regime	$F1_{NAO}$
Proposed work (Case I)	1-D CNN (single-lead, fixed-length)	90/10 split (held out)	0.756
Proposed work (Case II)	1-D CNN + BN (shallower conv stack)	90/10 split (held out)	0.710
Proposed work (Case III)	1-D CNN (uniform max-pool)	90/10 split (held out)	0.680
Zihlmann et al. (2017)	CNN/GRU, end-to-end	5-fold CV	0.792
Warrick & Homsi (2017)	CNN + LSTM, end-to-end	10-fold CV	0.83 ± 0.02
Zabihi et al. (2017)	491 hand-crafted features + RF	10-fold CV	0.819 ± 0.02
Hong et al. (2017)	Deep models + expert features (ensemble)	repeated CV	up to 0.866

From the table, *Case I* ($F1_{NAO} = 0.756$, accuracy = 0.815) establishes a credible single-model, single-lead baseline that is intentionally free of handcrafted features, ensembling, or sequence modules. The fact that it trails heavier systems) and ensemble/feature-hybrid methods is consistent with their additional capacity to model long-range temporal dependencies or to fuse heterogeneous feature views. By contrast, our architecture deliberately emphasizes compactness and

reproducibility, an end-to-end learning from fixed-length segments, and minimal preprocessing.

The per-class behavior (from our confusion matrices) explains much of the gap. All three configurations retrieve Normal with high recall, but the "Other" class remains the principal source of error due to its heterogeneity, and the Noise class (~) is sensitive to pooling schedules. *Case I's* kernel schedule (55/25/10/5) with global average pooling yields the strongest balance across N/A/O, whereas *Case II* (with batch normalization) improves robustness to noise at some cost to O-recall, and *Case III's* uniform pooling compresses away minority patterns. These are standard bias–variance trade-offs which the sequence models and ensembles mitigate by adding temporal context and model diversity, respectively.

However, it is noteworthy that the proposed model achieves competitive internal-validation performance without the complexity of feature engineering or assembling, making it suitable as a deployable patient-specific backbone where transparency, low latency, and straightforward retraining are valued. Industrially, the proposed model is light and computationally efficient hence making it appropriate to be incorporated in mobile health applications and ECG wearable devices. Compact models of this kind are of special interest in resource-constrained healthcare settings, in which high-throughput or cloud-based computation cannot be easily accessed. This ability to provide real-time arrhythmia detection on low power platforms highlights the practical value of such a solution, which is a way of providing scaled clinical decision support and, ultimately, constant patient monitoring.

Conclusion

We have introduced a small end-to-end, one-dimensional convolutional network to patient-specific classification of ECG rhythms in single-lead PhysioNet/CinC 2017. This model works on fixed length raw fragments with no use of handcrafted features or beat detectors. The wide to narrow kernel schedule with global average pooling produced the best balance between local morphology and the larger context rhythm over three architectural variants. In experimental analysis, we found that the recall of the Normal class is consistently high, with the remaining errors mostly occurring in heterogeneous other rhythms and noisy pieces, which are reflected in the class imbalance and boundary cases in single-lead records. The presented solution is purposely sparse, with only one model, insignificant preprocessing, clear hyperparameters, and consequently reproducible, easy to deploy, and competitive at the level of in-house validation. Although the suggested framework proves to be competitive

with a lean architecture, a number of constraints still exist including the need of dataset diversity, more rigorous validation procedures, and hyperparameter tuning etc. In future the framework has to extended to multi-lead and multi-center data, add higher levels of optimization and adaptive learning, and consider the possibility of using hybrid CNN-attention models, which maintain cost-efficiency but achieve a better representation of time. These instructions will also enhance the relevance of the suggested method in the clinical and industrial practices.

References

Ahmed, I. Khan, A. (2023). Learning based Compressive Speech Subsampling, *Multimed Tools Appl* 82, 15327–15343, Springer.

Ahmed, I., Khan, A. (2022) Genetic algorithm based framework for optimized sensing matrix design in compressed sensing. *Multimed Tools Appl* 81, 39077–39102.

Asl, B. M., Setarehdan, S. K., & Mohebbi, M. (2008). Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. *Artificial Intelligence in Medicine, 44*(1), 51–64. Elsevier.

Bammara, A. A., & MousseImal, A. (2025). *Classification of ECG signals using 1D-2D transformation and convolutional neural networks (CNN)* (Doctoral dissertation). Faculté des sciences et de la technologie.

Ding, C., Yao, T., Wu, C., & Ni, J. (2024). Deep learning for personalized electrocardiogram diagnosis: A review. *arXiv preprint arXiv:2409.07975*.

Ding, C., Yao, T., Wu, C., & Ni, J. (2025). Advances in deep learning for personalized ECG diagnostics: A systematic review addressing inter-patient variability and generalization constraints. *Biosensors and Bioelectronics, 271*, 117073. Elsevier.

Ikram, S., Ikram, A., Singh, H., Ali, D., Naveed, S., De La Torre, I., Gongora, H. F., & Candelaria, T. (2025). Transformer-based ECG classification for early detection of cardiac arrhythmias. *Frontiers in Medicine, 12*, 1600855. Frontiers.

Jin, J. (2018). Screening for cardiovascular disease risk with ECG. *JAMA, 319*(22), 2346. American Medical Association.

Joshi, A. J., Chandran, S., Jayaraman, V. K., & Kulkarni, B. D. (2009). Hybrid SVM for multiclass arrhythmia classification. *2009 IEEE International Conference on Bioinformatics and Biomedicine*, 287–290. IEEE.

Kalatehjari, E., Hosseini, M. M., Harimi, A., & Abolghasemi, V. (2025). Advanced ensemble learning-based CNN-BiLSTM network for cardiovascular disease classification using ECG and PCG signal. *Biomedical Signal Processing and Control, 108*, 107846. Elsevier.

Ma, L. Y., Chen, W. W., Gao, R. L., Liu, L. S., Zhu, M. L., Wang, Y. J., Wu, Z. S., Li, H. J., Gu, D. F., Yang, Y. J., & others. (2020). China cardiovascular diseases report 2018: An updated summary. *Journal of Geriatric Cardiology, 17*(1), 1.

Mahajan, M., Kadam, S., Kulkarni, V., Gujar, J., Naik, S., Bibikar, S., Ochani, A., & Pratap, S. (2024). ECG signal classification via ensemble learning: Addressing intra and inter-patient variations. *International Journal of Information Technology, 16*(8), 4931–

Perloff, J. K., & Marelli, A. (2012). *Clinical recognition of congenital heart disease* (Expert consult ed.). Philadelphia: Elsevier Health Sciences.

Samant, E. A. S. (2023). Exploring ECG signal analysis techniques for arrhythmia detection: A review. *International Journal of Recent Innovations in Trends in Computer Communication, 11*(9), 4881–4896.

Sulthana, A. R., & Jaithunbi, A. K. (2022). Varying combination of feature extraction and modified support vector machines-based prediction of myocardial infarction. *Evolving Systems, 13*(6), 777–794. Springer.

van der Bijl, K., Elgendi, M., & Menon, C. (2022). Automatic ECG quality assessment techniques: A systematic review. *Diagnostics, 12*(11), 2578. MDPI.

Wagner, G. S., & Strauss, D. G. (2014). *Marriott's practical electrocardiography*. Philadelphia: Lippincott Williams & Wilkins.

Zabihi, M., Rad, A. B., Katsaggelos, A. K., Kiranyaz, S., Narkilahti, S., & Gabbouj, M. (2017). Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier. *2017 Computing in Cardiology (CinC)*, 1–4. IEEE.

Zahid, M. U., Kiranyaz, S., & Gabbouj, M. (2022). Global ECG classification by self-operational neural networks with feature injection. *IEEE Transactions on Biomedical Engineering, 70*(1), 205–215. IEEE.