

Real-time phonocardiogram anomaly detection by adaptive 1D Convolutional Neural Networks



Serkan Kiranyaz^{a,*}, Morteza Zabihi^b, Ali Bahrami Rad^c, Turker Ince^d, Ridha Hamila^a, Moncef Gabbouj^b

^a Electrical Engineering Department, College of Engineering, Qatar University, Doha 2713, Qatar

^b Department of Computing Sciences, Tampere University, Tampere 35330, Finland

^c Department of Biomedical Informatics, Emory University, Atlanta, USA

^d Electrical & Electronics Engineering Department, Izmir University of Economics, Balçova, Izmir, Turkey

ARTICLE INFO

Article history:

Received 1 March 2019

Revised 29 December 2019

Accepted 19 May 2020

Available online 30 May 2020

Communicated by Zhu Yu

Keywords:

Phonocardiogram classification
Convolutional Neural Networks
Real-time heart sound monitoring

ABSTRACT

The heart sound signals (Phonocardiogram – PCG) enable the earliest monitoring to detect a potential cardiovascular pathology and have recently become a crucial tool as a diagnostic test in outpatient monitoring to assess heart hemodynamic status. The need for an automated and accurate anomaly detection method for PCG has thus become imminent. To determine the state-of-the-art PCG classification algorithm, 48 international teams competed in the PhysioNet (CinC) Challenge in 2016 over the largest benchmark dataset with 3126 records with the classification outputs, normal (N), abnormal (A) and unsure – too noisy (U). In this study, our aim is to push this frontier further; however, we focus deliberately on the anomaly detection problem while assuming a reasonably high Signal-to-Noise Ratio (SNR) on the records. By using 1D Convolutional Neural Networks trained with a novel data purification approach, we aim to achieve the highest detection performance and real-time processing ability with significantly lower delay and computational complexity. The experimental results over the high-quality subset of the same benchmark dataset show that the proposed approach achieves both objectives. Furthermore, our findings reveal the fact that further improvements indeed require a personalized (patient-specific) approach to avoid major drawbacks of a global PCG classification approach.

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

According to the World Health Organization (WHO) report in 2015, an estimated 17.7 million people died from cardiovascular diseases only in 2015, representing 37% of all premature deaths worldwide. The most common, and possibly the cheapest and the earliest clinical examination is heart auscultation, which can reveal several cardiac anomalies, such as certain arrhythmias, failures, murmurs, as well as diseases such as ventricular septal defects, and stenosis in the aorta [28] and many more. However, there are several limitations and practical problems in the human auditory system when it comes to phonocardiogram (PCG) signal analysis, despite the cognitive skills and expertise of the medical examiner. This is worsened by surrounding environmental noise and possibly large signal variations in the local recording areas. This brought the need for an automated, cost-effective and robust anomaly detection method for PCG signals. During the last 20 years,

several approaches [9,5,29,41,15]) have been proposed in the literature. However, most of them were tested on limited PCG datasets usually containing only a small number of records, e.g., a few dozens. Most of them used a particular machine learning approach, and it is a well-known fact that they can be easily tuned to maximize the classification performance on a limited test set. Moreover, it is not feasible to perform any comparative evaluation between them due to the variations of the train and test datasets used. The need for a benchmark dataset which contains a sufficiently large collection of PCG records was imminent, and finally, the PhysioNet community [30] composed the largest PCG dataset that was used as the benchmark dataset in PhysioNet (CinC) Challenge 2016, where 48 proposals have recently competed. The benchmark dataset contains 3126 records; however, some records were too difficult to analyze due to the significant noise; therefore, the classification outputs as, normal (N), abnormal (A) and noisy (U) were recommended by the Challenge. As a result, the objective of the competition is not only to detect records with anomalies (A) but also to detect those records which are “too noisy” (U). This was, therefore, a blended objective rather than a sole anomaly detection.

* Corresponding author.

E-mail addresses: mkiranyaz@qu.edu.qa (S. Kiranyaz), Morteza.Zabihi@tuni.fi (M. Zabihi), turker.ince@izmirekonomi.edu.tr (T. Ince), hamila@qu.edu.qa (R. Hamila), Moncef.gabbouj@tuni.fi (M. Gabbouj).

Our earlier work, Zabihi et al. [40], was the 2nd ranked method in the PhysioNet (CinC) Challenge, with respect to this combined objective, was indeed the best-performing method for PCG anomaly detection. So taking contribution Zabihi et al. [40] of the challenge as the reference, the three main objectives of this study are to: 1) further improve the state-of-the-art anomaly detection performance 2) reduce the false alarms (misclassification of normal records as abnormal), and 3) achieve a real-time processing ability with significantly lower delay and computational complexity. The detection of low-quality (too noisy) records is beyond the scope of this study and therefore, we shall assume that each PCG records has a sufficiently high Signal-to-Noise Ratio (SNR) and can be segmented with a reasonably high accuracy. SNR is defined as the ratio of the signal power to noise power. Specifically, $SNR = 10 \log \left(\frac{\sigma_{signal}^2}{\sigma_{noise}^2} \right)$. The latter assumption has already been fulfilled by several recent methods (i.e., >95% accuracy is achieved in [29,31]). The current state-of-the-art method in Zabihi et al. [40] is also immune to PCG segmentation errors since it classifies the entire PCG record as a whole. However, the method also presents a high computational complexity due to the various hand-crafted feature extraction techniques employed. Hence the real-time processing may not be feasible, not only because of the such high computational complexity but also due to the fact that the entire PCG record should be acquired first.

Deep Convolutional Neural Networks (CNNs) are feed-forward artificial neural networks which were initially developed as the crude models of the mammalian visual cortex. Deep CNNs have recently become the *de-facto* standard for many visual recognition applications (e.g., object recognition, segmentation, tracking, etc.) as they achieved the state-of-the-art performance [6,32,22] with a significant performance gap. Other deep networks such as Deep Recurrent Neural Nets and Deep Belief Networks and deep CNNs have been used in recent applications in healthcare [37,38,10]. Recently, 1D CNNs have been proposed for pattern recognition for 1D signals such as patient-specific ECG classification [18,19] structural health monitoring [1,2,4] and mechanical and motor fault detection systems, [8,3,13,16]. Because there are numerous advantages of using a compact 1D CNN instead of a conventional (2D) deep counterpart. First of all, compact 1D CNNs can be efficiently trained with a limited dataset in 1D (e.g., [18,19,1,2,4,8,3,13]) while the 2D deep CNNs, besides the 1D to 2D data transformation, require datasets with massive size, e.g., in the “Big Data” scale in order to prevent the well-known “Overfitting” problem. In fact, this requirement alone makes deep 2D CNNs inapplicable to many practical problems that have limited datasets, including the problem addressed in this study. The crucial advantage of the CNNs is that both feature extraction and classification operations are fused into a single machine learning body to be jointly optimized to maximize the classification performance. This eliminates the need for hand-crafted features or any other post processing.

In order to address the aforementioned drawbacks, in this article we propose a novel PCG anomaly detection scheme that employs a compact 1D CNN and a data purification process that is embedded into the Back-Propagation (BP) training algorithm. Due to the simple structure of the 1D CNNs that requires only 1D convolutions (scalar multiplications and additions), a real-time and low-cost hardware implementation of the proposed approach is quite feasible. Finally, it is a well-known fact that the abnormal PCG records may and usually do have a certain amount of normal sound beats and hence they will inevitably cause a certain level of learning confusion which eventually leads to a degradation on the classification performance. The data purification that is embedded into the BP training is designed to reduce this during the training phase (offline) without causing any further time delay on the

detection. Overall, the proposed system has the following innovative and novel contributions:

- 1) It improves the state-of-the-art anomaly detection performance by Zabihi et al. [40] achieved in PhysioNet (CinC) Challenge.
- 2) It reduces false alarms (misclassification of normal records as abnormal).
- 3) It achieves a real-time processing ability with significantly lower delay and computational complexity.
- 4) Finally, it discusses the major limitation of the PCG classification problem in general and suggests a solution to further improve the detection performance.

The rest of the paper is organized as follows: Section II first presents the previous work in PCG classification with the main challenges tackled. Then PhysioNet/CinC Challenge will be presented and the focus is particularly drawn on the best method [40] of the challenge that achieves the highest score in PCG anomaly detection. The proposed anomaly detection approach using a compact 1D CNN is presented in Section III. In Section IV, extensive comparative evaluations against Zabihi et al. [40] will be performed using the standard performance metrics over the two datasets each of which is created from the PhysioNet/CinC benchmark database. Finally, Section VI concludes the paper and suggests topics for future research.

2. Prior work

2.1. Overview

Heart auscultation is performed on the anterior chest wall and used as the primary diagnostic method to evaluate heart function. A PCG signal shows the auscultation of turbulent blood flow and the timing of the heart valves' movements. The main components of a PCG signal in each heartbeat are S1 (first) and S2 (second) sounds, which are related to the closure of mitral and tricuspid valves, and the closure of aortic and pulmonic valves, respectively.

Hearing abnormal heart sounds, such as transients and murmurs, in a PCG signal is usually considered pathological. These abnormal sounds can provide information about the blockage of the valve when it is open (stenosis), the leakage of the valve when it is closed (insufficiency), the increase of blood flow, etc. However, the correct interpretation of such signals depends on the acuity of hearing and the experience of the observer.

Several studies have been presented to automatically identify PCG anomalies using signal processing and machine learning techniques. There are two types of PCG signal analysis and anomaly detection methods. The first type analyzes the entire PCG record without segmentation [39,7] whereas the second type uses temporal beat segmentation, i.e. identifying the cardiac cycles and localizing the position of the first (S1; beginning of the systole) and second (S2; end of the systole) primary heart sounds. Certain variations over S1 and S2 properties, such as their duration or intensities, can be considered as the primal signs of cardiac anomalies. For PCG segmentation (or beat detection) there exist various prior works using different envelope extraction methods such as Shannon energy [9], Shannon entropy [29], Hilbert-Huang transform [41], and autocorrelation [15]. Some methods use envelope extraction based on wavelet transform to gain the frequency characteristics of S1 and S2 sound [11].

PCG classification is especially challenging due to high variations among the N and A type PCG patterns and the noise levels. Fig. 1 shows some typical PCG beats where different records in the benchmark Physionet/CinC dataset [24] have entirely different

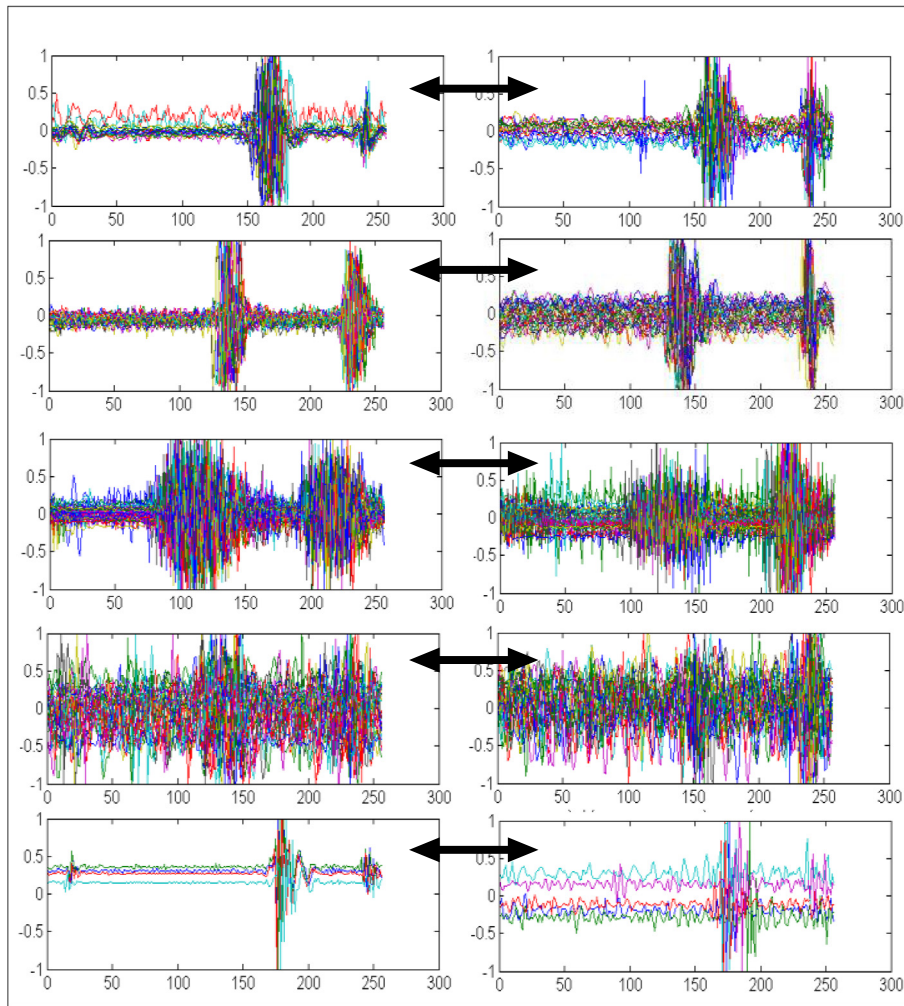


Fig. 1. N (left) vs. A (right) beats from different subject records in Physionet/CinC dataset [30].

N beats, which may, however, show a high level of structural similarities to other A beats (e.g. see the pairs with arrows in the figure). Obviously, past approaches that relied on only one or few hand-crafted features may not characterize all such inter- and intra-class variations and thus they have usually exhibited a common drawback of having an inconsistent performance when, for instance, classifying a new patient's PCG signal. This makes them unreliable for wide clinical use.

2.2. PhysioNet/CinC challenge

For this challenge, 3126 PCG labelled records (including training and validation sets) were provided by Physionet/Computing in Cardiology Challenge 2016 [30]. The provided database was sourced by several international contributors. The database was collected from both healthy and pathological subjects in clinical and non-clinical environments. In the database, each PCG record is labeled as N or A. In addition, the quality of each recording was also identified either as good or bad. The task was to design a model that can automatically classify the PCG recordings as N, A, or too noisy to evaluate. More detailed information can also be found in Liu et al. [24].

2.3. The best anomaly detection method: Zabihi et al. [40]

As illustrated in Fig. 2, a set of 18 features was selected from a comprehensive set of hand-crafted features using a greedy

wrapper-based feature selection algorithm. The selected features consisted of the linear predictive coefficients, natural and Tsallis entropies, mel frequency cepstral coefficients, wavelet transform coefficients, and power spectral density functions [40]. Then, the features were classified through a two-step algorithm. In the first step, the quality of PCG recordings was detected, and in the second step, the PCG records with good quality were further classified into N or A classes.

The classification algorithm in Zabihi et al. [40] consists of an ensemble of 20 feedforward sigmoidal Neural Networks (NNs) with two hidden layers, and 25 hidden neurons in each layer. The output layer has 4 neurons to detect the signal quality (good vs. bad) and anomaly (N vs. A), simultaneously. This ensemble of NNs was trained via a 20-fold cross-validation committee with Bayesian regularization back-propagation training algorithm [26], and a bootstrap resampling method to make the size of the N and A classes balanced. The final classification rule which was learned via a 10-fold cross-validation method, combines the decision of all 20 NNs such that if at least 17 NNs classified a signal as bad quality, the algorithm recognized it as bad quality, otherwise it is detected as good quality. Good quality signals were further classified as A if at least 7 NNs recognized it as A, otherwise, it was detected as N.

Since the objective function (i.e., the evaluation mechanism) and the dataset of the PhysioNet (CinC) Challenge were different from the current study, we adapted the method in Zabihi et al. [40], i.e., in the output layer of ANNs we now used two neurons

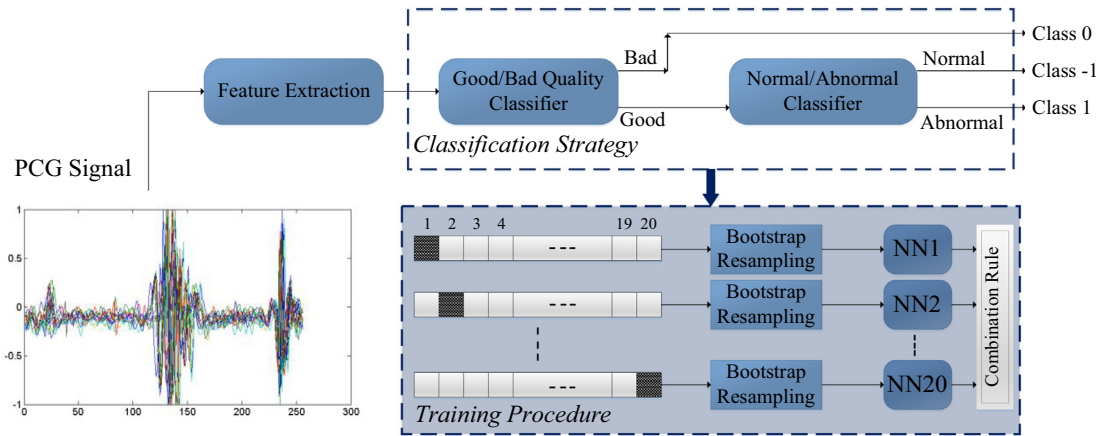


Fig. 2. Schematic demonstration of classification strategy and training procedure of Zabihi et al. [40].

rather than four since we do not need to detect the quality of PCG signals anymore.

3. The proposed approach

Here we propose a novel approach for real-time monitoring heart sound signals for robust and accurate anomaly detection. To accomplish the aforementioned objectives, we used a 1D CNN at the core of the system that is trained by labeled and segmented PCG records by the PhysioNet (CinC) Challenge organizers. Data purification is periodically performed during the BP training in order to reduce the confusion due to the potential occurrence of normal (N) beats in abnormal (A) records. Once the 1D CNN is properly trained, it can classify each monitored PCG beat in real-time, or alternatively classify a PCG record of a patient as N or A.

As illustrated in Fig. 3, the PCG signal can be acquired either from a PCG monitoring device in real-time or retrieved from a PCG file. The data processing block first segments the PCG stream into PCG beats. Each PCG beat, p , is linearly normalized in the range of $[-1, 1]$ as follows:

$$p_i \leftarrow 2 \frac{p_i - \min(p)}{\max(p) - \min(p)} - 1 \quad (1)$$

where p_i is the i th sample in the PCG beat, p . Eq. (1) linearly maps the minimum and maximum samples to -1 and 1 , respectively, and hence negates the effect of the sample amplitudes in the learning process. Each normalized beat is then fed into the 1D CNN classifier that has already been trained as shown in Fig. 4. For a real-time

operation the output of the 1D CNN can be used directly or alternatively can be processed by a majority rule to obtain the final class decision of the entire stream, e.g., the stream is classified as A if more than 25% of the beats are abnormal. This threshold should be determined in a practical way so that it should be sufficiently high to prevent false-alarms due to classification noise while not too high to detect those abnormal records with few abnormal PCG beats. Therefore, the “Final Decision” block determines the final class type of a PCG file while its utilization is optional for real-time PCG anomaly detection.

In the rest of this section, an overview of 1D CNNs will first be presented and the BP training with data purification will be detailed next.

3.1. 1D CNN overview

CNNs are a type of feed-forward artificial neural networks and they are mainly used for 2D signals such as images and video [6,32,22]. In this study, we use compact 1D CNNs in order to accomplish the objectives presented in Section I. There are two types of layers in 1D CNNs: 1) CNN-layers where both 1D convolutions and sub-sampling occur, and 2) Fully-connected layers that are identical to the hidden and output layers of a typical Multi-Layer Perceptron (MLP). Fig. 5 illustrates the three consecutive CNN layers with the kernel size 3 and sub-sampling factor, $SS = 2$. These network parameters are chosen for the illustrative purposes. The left-most layer ($l - 1$), the output feature map (array) size is 22 and next layer (l) input array size becomes $22 - (3 - 1) = 20$ when 1D convolution is performed. The output

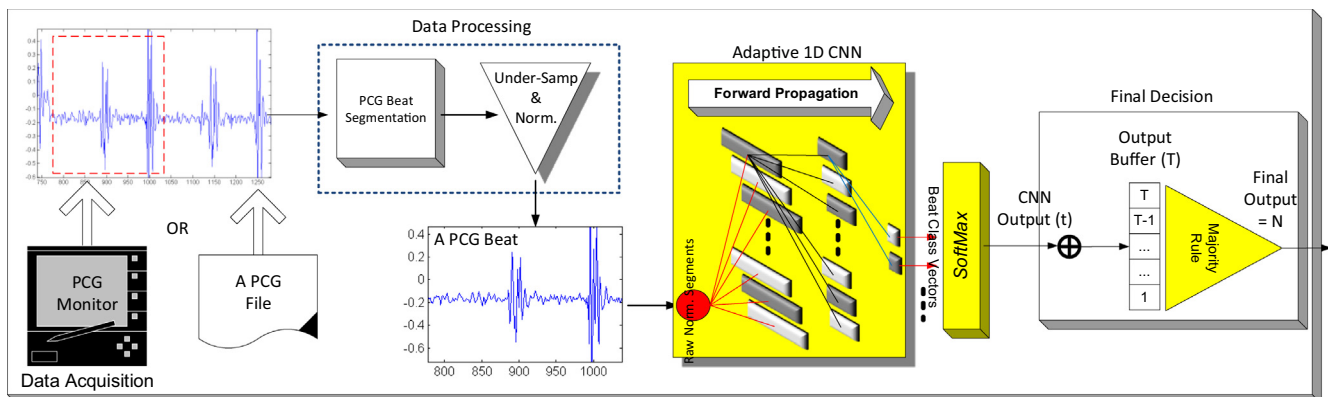


Fig. 3. The proposed system architecture for real-time PCG anomaly detection after the BP training of the 1D CNN as illustrated in Fig. 4.

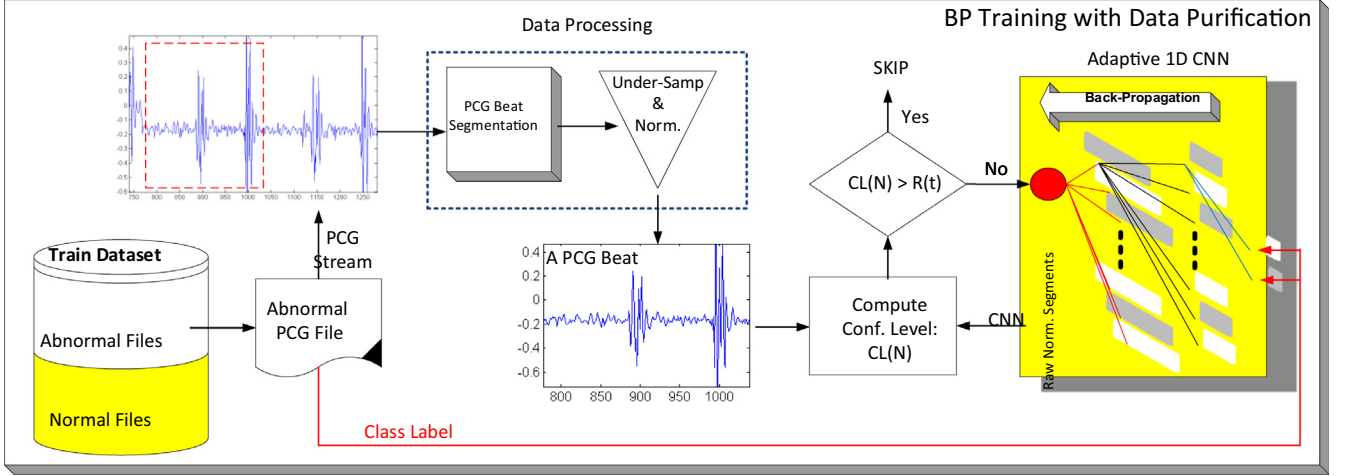


Fig. 4. The (offline) BP training of the 1D CNN for abnormal PCG beats.

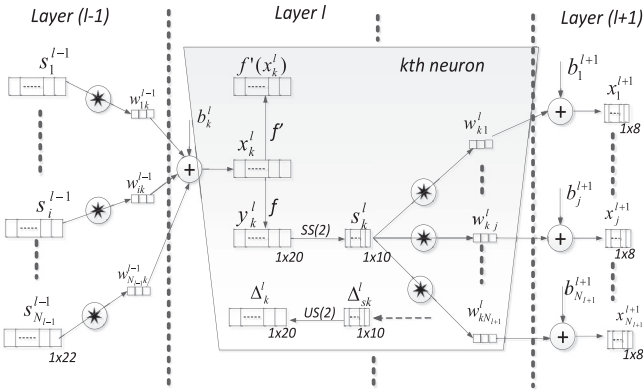


Fig. 5. Three CNN layers of a 1D CNN [13].

feature array size will then be 10 with a subsampling of a factor 2. After another convolution with a kernel of size 3, the input feature array at layer $l + 1$ will finally have a size of $10 - (3 - 1) = 8$. In order to exhibit the adaptive nature of the proposed 1D CNN, assume that the layer $l + 1$ is the output (last) CNN layer. In this case the sub-sampling factor in this layer will *adaptively* set to 8 to create the output array size of 1 (scalar) that can then be connected to the dense (MLP) layers directly. This will, therefore, ensure that the output size will always be 1 (scalar) regardless of the number of CNN layers and the input signal (PCG segment) duration. With this approach, it is aimed that the network parameters, i.e., the number of CNN layers, sub-sampling factors and kernel sizes can be set freely without any restrictions.

In the proposed approach 1D CNNs have a simple and shallow configuration with only a few hidden layers and neurons. Typically, number of parameters in the network is only few hundreds and the network can adapt itself to the duration (resolution) of each PCG segment. In the CNN-layers, the 1D forward propagation (FP) between the two CNN layer can be expressed as,

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} \text{conv1D}(w_{ik}^{l-1}, s_i^{l-1}) \quad (2)$$

where x_k^l is the input, b_k^l is the bias of the k th neuron at layer l , and s_i^{l-1} is the output of the i th neuron at layer $l - 1$. w_{ik}^{l-1} is the kernel from the i th neuron at layer $l - 1$ to the k th neuron at layer l .

3.2. Back-propagation with data purification

We shall now briefly formulate the back-propagation (BP) steps while skipping the detailed derivations as further details can be found in Kiranyaz et al. [18]. The BP of the error starts from the MLP output layer. Let $l = 1$ and $l = L$ be the input and output layers, respectively. The (empirical) error in the output layer can be written as,

$$E = E(y_1^l, y_2^l) = \sum_{i=1}^2 (y_i^l - t_i)^2 \quad (3)$$

For an input vector p , and its corresponding output and target vectors, $[y_1^l, y_2^l]$ and $[t_1^l, t_2^l]$, respectively, we aim at finding out the derivative of this error with respect to an individual weight (connected to that neuron, k) w_{ik}^{l-1} , and bias of the neuron k , b_k^l , so that we can perform gradient descent method to minimize the error accordingly. Once all the delta errors in each MLP layer are determined by the BP, then weights and bias of each neuron can be updated by the gradient descent method. Specifically, the delta error of the k th neuron at layer l , Δ_k^l , will be used to update the bias of that neuron and all weights of the neurons in the previous layer connected to that neuron, as:

$$\frac{\partial E}{\partial w_{ik}^{l-1}} = \Delta_k^l y_i^{l-1} \quad \text{and} \quad \frac{\partial E}{\partial b_k^l} = \Delta_k^l \quad (4)$$

So from the input MLP layer to the output CNN layer, the regular (scalar) BP is simply performed as,

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} w_{ki}^l \quad (5)$$

Once the first BP is performed from the next layer, $l + 1$, to the current layer, l , then we can further back-propagate it to the input delta, Δ_k^l . Let zero order up-sampled map be: $us_k^l = up(s_k^l)$, then one can write:

$$\Delta_k^l = \frac{\partial E}{\partial y_k^l} \frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial E}{\partial us_k^l} \frac{\partial us_k^l}{\partial y_k^l} f'(x_k^l) = up(\Delta s_k^l) \beta f'(x_k^l) \quad (6)$$

where $\beta = (ss)^{-1}$ since each element of s_k^l was obtained by averaging ss number of elements of the intermediate output, y_k^l . The inter-BP (among CNN layers) of the delta error ($\Delta s_k^l \xrightarrow{\Sigma} \Delta_{A \pm}^{l+1}$) can be expressed as,

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} \text{conv1Dz}(\Delta_i^{l+1}, rev(w_{ki}^l)) \quad (7)$$

where $rev(\cdot)$ reverses the array and $conv1Dz(\cdot)$ performs full convolution in 1D with $K-1$ zero padding. Finally, the weight and bias sensitivities can be expressed as,

$$\frac{\partial E}{\partial w_{ki}^l} = conv1D(s_k^l, \Delta_i^{l+1}) \frac{\partial E}{\partial b_k^l} = \sum_n \Delta_k^l(n) \quad (8)$$

3.2.1 Back-propagation flowchart

As a result, the iterative flow of the BP for the N beats in normal PCG records in the training set can be stated as follows:

- 1) Initialize weights and biases (usually randomly, $U(-a, a)$) of the network.
- 2) For each BP iteration DO:
 - a. For each PCG beat in the dataset, DO:
 - i. **FP**: Forward propagate from the input layer to the output layer to find outputs of each neuron at each layer, $y_i^l, \forall i \in [1, N_l]$ and $\forall l \in [1, L]$.
 - ii. **BP**: Compute delta error at the output layer and back-propagate it to first hidden layer to compute the delta errors, $\Delta_k^l, \forall k \in [1, N_l]$ and $\forall l \in [2, L-1]$.
 - iii. **PP**: Post-process to compute the weight and bias sensitivities using Eq. (8)
 - iv. **Update**: Update the weights and biases with the (cumulation of) sensitivities found in (c) scaled with the learning factor, ε :

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \varepsilon \frac{\partial E}{\partial w_{ik}^{l-1}} \quad (9)$$

$$b_k^l(t+1) = b_k^l(t) - \varepsilon \frac{\partial E}{\partial b_k^l}$$

3.2.2. Data purification

The proposed method works over the PCG beats (frames) while the labeling information is for the PCG records (files) of the benchmark dataset created by the CinC competition. So when a PCG record is labeled as N , it is clear that all the PCG beats in this record are normal. However, if a PCG record is labeled as A , this does not necessarily mean that all the PCG beats are abnormal. In this case, there are certain amount of normal beats in this abnormal record and it is quite possible that these normal beats can be even the majority of this record. Therefore, while training the 1D CNN, these normal beats should not be included in the training with a label “ A ” because 1D CNN will obviously “confuse” while processing a normal beat having the label “ A ”. To avoid this confusion we altered the conventional BP process for back-propagating the error from each abnormal PCG beat from an abnormal file. As illustrated in Fig. 4, during each BP iteration, the error from a PCG beat in an abnormal record will be back-propagated if the current CNN does not classify it as a N beat with a high confidence level. In other words, if the current CNN classifies it as a N beat with a sufficiently high confidence level, i.e., $CL(N) > R(t)$ where t is the current BP iteration, then it will be dropped out of the training process until the next periodic check-up. The dynamic threshold, $R(t)$, for the confidence level is gradually reduced since the CNN can distinguish the N beats with a higher accuracy as the BP training advances and thus during the advanced stages of the BP those beats that are classified as N with a reasonable confidence level can now be conveniently skipped from the training. Note that the aim of this data purification is to reduce a potential confusion (e.g., N beats in an abnormal PCG record). An alternative action would be to reverse their truth labels (A) to N and still include them in the BP training. However, this is only an “expected” outcome indicated by a partially trained CNN, not a proven one. Hence such an approach

may cause a possible error accumulation. Even if some of those beats are still abnormal despite the CNN classification, skipping them from the training may only reduce the generalization capability of the classifier due to reduced training data.

Accordingly, the **FP** and **BP** operations in the 2nd BP step can be modified as follows:

2) For each BP iteration, t , DO:

a) For each PCG beat in an abnormal record, DO:

- i. **FP**: Forward propagate from the input layer to the output layer to find outputs of each neuron at each layer, $y_i^l, \forall i \in [1, N_l]$ and $\forall l \in [1, L]$.
- ii. **Periodic Check (Pc)**: At every P_c iteration DO:

Compute: $CL(N) = 50(y_1^l - y_2^l)$. If $CL(N) > R(t)$, mark this beat as “skip”.

- iii. **Skip**: If the beat is marked as “skip”, drop this beat out of BP training and proceed with the next beat.
- iv. **BP**: Compute delta error at the output layer and back-propagate it to first hidden layer to compute the delta errors, $\Delta_k^l, \forall k \in [1, N_l]$ and $\forall l \in [2, L-1]$.

4. Experimental results

In this section, the experimental setup for the test and evaluation of the proposed systematic approach for PCG anomaly detection will be first presented. Then an evaluation of the data purification process will be visually performed over two PCG sample records and the results will be discussed. Next using the conventional performance metrics, the overall abnormal beat detection performance and the robustness of the proposed system against variations of the system and network parameters will be evaluated. Finally, the computational complexity of the proposed method for both offline and real-time operations will be reported in detail.

4.1. Experimental setup

In the PCG dataset created in the Challenge, the sampling rate is 2 kHz. In a normal rhythm, one can expect around 1 beat per second. In a paced activity, this can go up to maximum 4 beats/second. So, the number of samples in a beat will vary between 500 and 2000 samples. In order to feed the raw PCG signal as the input frame to 1D CNN, the frame size should be fixed to a constant value. Therefore, in this study each segmented PCG beat has down-sampled to 1000 samples, which is a practically high value to capture each beat with a sufficiently high resolution and allows the network to analyze the beat characteristics independent from its duration. We used a compact 1D CNN in all experiments with only 3 hidden CNN layers and 2 hidden MLP layers, in order to achieve the utmost computational efficiency for both training and particularly for real-time anomaly detection. The 1D CNN used in all experiments has 24, neurons in all (hidden) CNN and MLP layers. The output (MLP) layer size is 2 as this is a binary classification problem and the input (CNN) layer size is 1. The kernel size of the CNN is 41 and the sub-sampling factor is 4. Accordingly, the sub-sampling factor for the last CNN-layer is automatically set to 10. Fig. 6 illustrates the compact network configuration used in the experiments.

In order to form the PCG dataset with a sufficiently high SNR, we estimated the noise variance of each record over the low signal activity (diastole) section of each PCG beat (e.g. the first 20% of each segment). Because when the signal activity is minimal, the variance computation will belong to noise and it will hence be

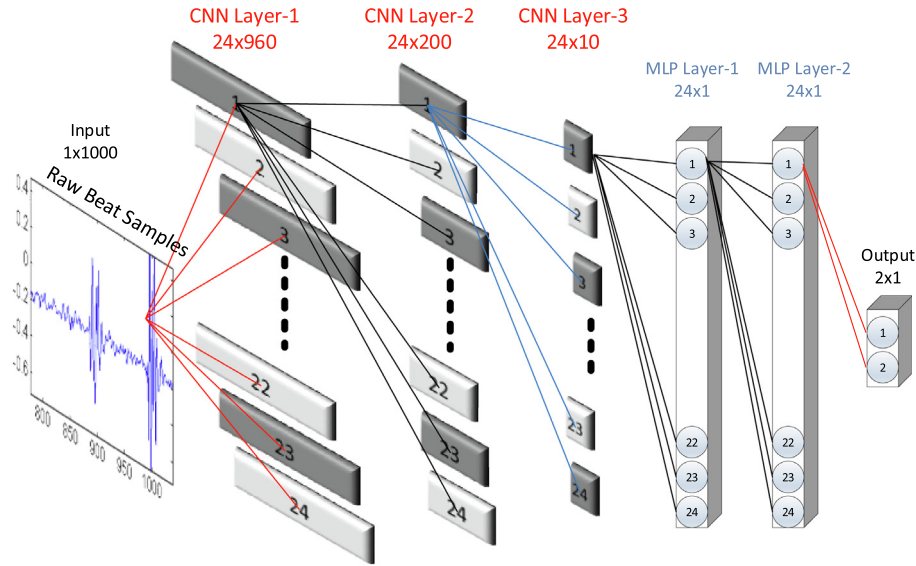


Fig. 6. 1D CNN configuration with 3 CNN and 2 MLP layers.

inversely proportional to the SNR. To compute the noise average of a PCG record we will then take the average of the beat variances. According to their average noise variances, we then sorted the PCG records in the database and then took the top 1200 PCG records (900 N and 300 A) to constitute the high-SNR PCG dataset. The estimated average SNR of the worst (bottom-ranked) PCG record in this list is around -6 dB, which means that the noise power (variance) is nearly 4 times higher than the signal power. In order to evaluate the noise effect over the proposed approach, we also form the 2nd dataset with the lowest possible SNR. For this purpose, we selected 1008 (756 N and 252 A) PCG records from the bottom of the sorted list. The estimated average SNR of the best (top-ranked) PCG record in this list is around -9 dB, which means that the noise power (variance) is about 8 times higher than the signal power. Over both high- and low-SNR datasets, we performed 4-fold cross validation and for each fold, one partition (25%) is kept for testing while the other three (75%) are used for training. This allows us to test the proposed system over all the records. Since the BP training algorithm is a gradient descent algorithm, which has a stochastic nature and thus can present varying performance levels, for each fold we performed 10 randomly initialized BP runs and computed their 2×2 confusion matrices that are then accumulated to compute the overall confusion matrix (CM). Each CM per fold is then accumulated to obtain the final CM that can be used to calculate the average performance metrics for an overall anomaly detection performance, i.e., classification accuracy (Acc), Sensitivity (Sen), Specificity (Spe), and Positive predictivity (Ppr). The definitions of these standard performance metrics using the hit/miss counters obtained from the CM elements such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN), are as follows: *Accuracy* is the ratio of the number of correctly classified beats to the total number of beats, $Acc = (TP + TN) / (TP + TN + FP + FN)$; *Sensitivity* (Recall) is the rate of correctly classified A beats among all A beats, $Sen = TP / (TP + FN)$; *Specificity* is the rate of correctly classified N beats among all N beats, $Spe = TN / (TN + FP)$; and *Positive predictivity* (Precision) is the rate of correctly classified A beats in all beats classified as A, $Ppr = TP / (TP + FP)$.

For all experiments we employ an early stopping training procedure with both setting the maximum number of BP iterations to 50 and the minimum training error to 8% to prevent overfitting. We use $Pc = 5$ iterations and initially set the learning factor, ϵ , as 10^{-3} . We applied a global adaptation during each BP iteration,

i.e., if the train MSE decreases in the current iteration we slightly increase ϵ by 5%; otherwise, we reduce it by 30%, for the next iteration.

4.2. Results and comparative evaluations

The experimental results with the data purification block have shown an average accuracy improvement around 1.8% on the test set and therefore, we can conclude that the proposed method can reduce the confusion caused by the potential N beats in abnormal records.

The comparative evaluations are performed over the PCG records in the test partitions of the 4-folds. It is worth mentioning that we used the competing method with minimum adaptation. To be more specific the competing method was designed for PCG quality and anomaly detection while in this work the focus is only on the anomaly detection. The threshold for the majority rule, Ta , is varied between $[0.1, 0.4]$. As discussed earlier, this threshold determines the minimum rate of A beats in a record to classify it as A. As Ta gets lower, it is more likely to detect higher number of actual abnormal records and hence the sensitivity (recall) rate would increase; however, as a trade-off, the false alarms, too, would also increase due to variations and noise effects in normal records. This, in turn, reduces both *Specificity* and *Positive Predictivity* (Precision). Depending on the target application, this threshold can therefore be set in order to maximize or favor either *Sensitivity* or *Specificity-Positive Predictivity*.

Over the HSNR dataset, Fig. 7 show the average Spe vs. Sen (top) and Ppr vs. Sen (bottom) plots with varying Ta values, respectively. The red circle in both plots indicates the corresponding performance values achieved by the competing method, i.e., $Sen = 0.8967$, $Spe = 0.8689$ and $Ppr = 0.6970$.

We can now evaluate the effect of such (high) noise presence over the proposed approach. For this purpose Fig. 9 shows the average Spe vs. Sen (top) and Ppr vs. Sen (bottom) plots with varying Ta values over the low-SNR dataset. Once again the red circle in both plots indicates the corresponding performance values achieved by the competing method, i.e., $Sen = 0.8135$, $Spe = 0.922$ and $Ppr = 0.7795$. Comparing with the results over high-SNR dataset, the high noise level degraded the *Sensitivity* levels of both methods. As an expected outcome, the degradation is worse for the proposed approach since it can only learn from the pattern of

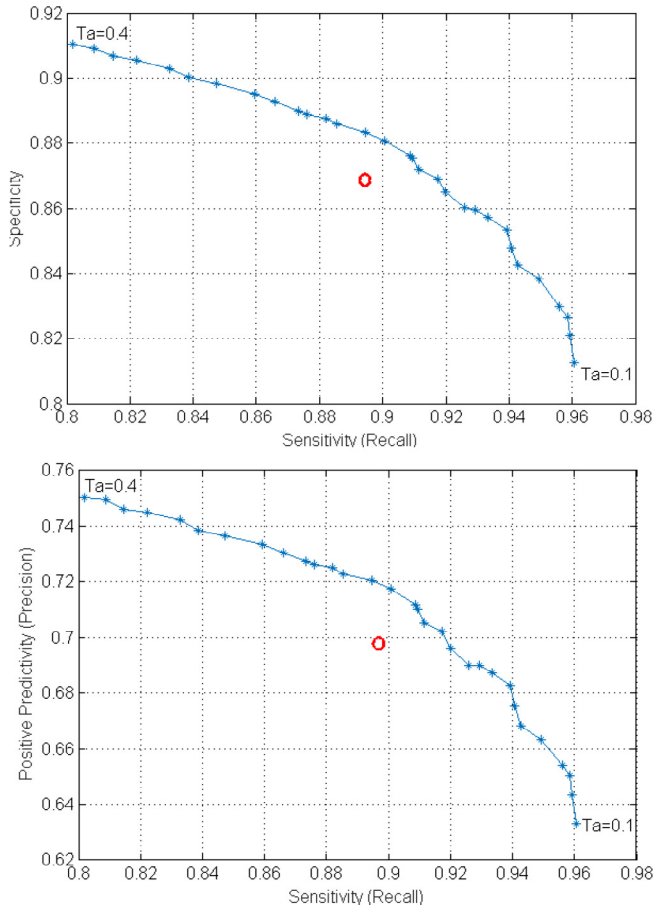


Fig. 7. Average Specificity (top) and Positive Predictivity (bottom) vs. Sensitivity (Recall) plots with varying T_a in the range [0.1, 0.4] for the test partitions of 4-fold cross validation over the high-SNR dataset. The red circle represents the performance of Zabihi et al. [40].

a single beat whereas the competing method can use several global features that are based on the joint characteristics and statistical measures of the entire record.

Finally, in order to evaluate the effects of hyper-parameter variations such as number of neurons per (hidden) layer and kernel size, we used two alternative 1D CNNs. In each network we change one hyper-parameter (either neuron number or kernel-size) while other hyper-parameters are kept intact. In the 1st configuration (C1), we increased the number of neurons 50%, i.e., 36 neurons per hidden (both CNN and MLP) layer. In the 2nd configuration, we reduced the kernel size by half (21). In this configuration, the sub-sampling factor for the last CNN-layer is automatically set to 36. We applied the same training approach (4-fold cross validation and 10 BP runs per fold) whilst using the identical train parameters. The 1st and 2nd network configurations yield 0.4% improvement and 0.16% reduction on the test accuracy level of the baseline configuration, respectively. Such insignificant performance variations show the robustness of the 1D CNNs and one can argue to increasing the number of neurons 50% will cause around 3-times increase in the computational complexity and hence it may not be desired to achieve such an insignificant accuracy gain.

4.3. Computational complexity

In order to analyze the computational complexity for both FP and BP process, we shall first compute the total number of opera-

tions at each 1D CNN layer (ignoring the sub-sampling) and then cumulate them to find the overall computational complexity. During the FP, at a CNN layer, l , the number of connections to the previous layer is, $N^{l-1}N^l$. the number of connections to the previous layer is, $N^{l-1}N^l$, there is an individual linear convolution is performed, which is a linear weighted sum. Let s^{l-1} and w^{l-1} and be the vector sizes of the previous layer output, s_k^{l-1} , and kernel (weight), w_{ki}^{l-1} , respectively. Ignoring the boundary conditions, a linear convolution consists of $s^{l-1}(w^{l-1})^2$ multiplications and s^{l-1} additions from a single connection. Ignoring the bias addition, the total number of multiplications and additions in the layer l will therefore be:

$$\begin{aligned} N(\text{mul})^l &= N^{l-1}N^l s^{l-1} (w^{l-1})^2, \\ N(\text{add})^l &= N^{l-1}N^l s^{l-1} \end{aligned} \quad (10)$$

So during FP the total number of multiplications and additions, $T(\text{mul})$, and $T(\text{add})$, on a L CNN layers will be,

$$\begin{aligned} T_{FP}(\text{mul}) &= \sum_{l=1}^L N^{l-1}N^l s^{l-1} (w^{l-1})^2, \\ T_{FP}(\text{add}) &= \sum_{l=1}^L N^{l-1}N^l s^{l-1} \end{aligned} \quad (11)$$

Obviously, $T(\text{add})$ is insignificant compared to $T(\text{mul})$.

During the BP, there are two convolutions performed as expressed in Eqs. (7) and (8). In Eq. (7), a linear convolution between the delta error in the next layer, Δ_i^{l+1} , and the reversed kernel, $rev(w_{ik}^l)$, in the current layer, l . Let x^l be the size of both the input, x_i^l , and also its delta error, Δ_i^l , vectors of the i th neuron. The number of connections between the two layers is, $N^{l+1}N^l$ and at each connection, the linear convolution in Eq. (7) consists of $x^{l+1}(w^l)^2$ multiplications and x^{l+1} additions. So, again ignoring the boundary conditions, during a BP iteration, the total number of multiplications and additions due to the first convolution will, therefore, be:

$$\begin{aligned} T_{BP}^1(\text{mul}) &= \sum_{l=0}^{L-1} N^{l+1}N^l s^{l-1} (w^{l-1})^2, \\ T_{BP}^1(\text{add}) &= \sum_{l=0}^{L-1} N^{l+1}N^l s^{l-1} \end{aligned} \quad (12)$$

The second convolution in Eq. (8) is between the current layer output, s_k^l , and next layer delta error, Δ_i^{l+1} where $w^l = x^{l+1} - s^l$. For each connection, the number of additions and multiplications will be, w^l and $w^l(x^{l+1})^2$, respectively. During a BP iteration, the total number of multiplications and additions due to the second convolution will, therefore, be:

$$\begin{aligned} T_{BP}^2(\text{mul}) &= \sum_{l=0}^{L-1} N^{l+1}N^l w^l (x^{l+1})^2, \\ T_{BP}^2(\text{add}) &= \sum_{l=0}^{L-1} N^{l+1}N^l w^l \end{aligned} \quad (13)$$

So at each BP iteration, the total number of multiplications and additions will be, $(T_{FP}(\text{mul}) + T_{BP}^1(\text{mul}) + T_{BP}^2(\text{mul}))$ and $(T_{FP}(\text{add}) + T_{BP}^1(\text{add}) + T_{BP}^2(\text{add}))$, respectively. Obviously, the latter is insignificant compared to former especially when the kernel size is high. Moreover, both operation complexities are proportional to the total number of connections between two consecutive layers, which are the multiplication of the number of neurons at each layer. Finally, the computational complexity analysis of MLPs is well known (e.g., see [33]) and it is quite negligible in the current implementation since only a scalar (weight) multiplication and an addition are performed for each connection.

The implementation of the 1D CNN is performed using C++ over MS Visual Studio 2013 in 64bit. This is a non-GPU implementation; however, Intel[®] OpenMP API is used to obtain multiprocessing with a shared memory. We used a 48-core Workstation with 128 Gb memory for the training of the 1D CNN. One expects 48× speed improvement compared to a single-CPU implementation; however, the speed improvement was observed to be around 36× to 39× in practise.

The average time for 1 BP iteration (1 consecutive FP and BP) per PCG beat was about 0.57 msec. A BP run typically takes 5 to 30 iterations over the training partition of the high-SNR dataset, which contains around 30,000 PCG beats overall. Therefore, the average time for training the 1D CNN classifier was around, $30000 * 30 * 0.57\text{msec} = 513 \text{ s} (<9 \text{ min})$ which is a negligible time since this is a one-time (offline) operation.

5. Discussion

A wide range of experimental results show that the proposed approach achieves superior performance levels with respect to all three metrics for a certain range of Ta values (i.e., $0.2 < Ta < 0.3$). Alternatively, setting $Ta < 0.15$ yields only around <0.04 less *Specificity* and *Positive predictivity* than the competing method; whilst the proposed approach can now achieve close to 0.96 *Sensitivity* level which basically indicates that except only few, almost all abnormal records are successfully detected.

Under the condition that a reasonable signal quality (SNR) is present, the experimental results show that all the objectives have been achieved and furthermore the proposed approach can conveniently be used in portable, low-power PCG monitors or acquisition devices for automatic and real-time anomaly detection. This is a unique advantage of the proposed approach over the best method [40] of the Physionet challenge, which can only classify an entire PCG record as a whole. On the other hand, for PCG file classification, the unique parameter, Ta , for majority rule decision can be set in order to favor either *Sensitivity* or *Specificity* – *Positive predictivity* (or both) depending on the application requirements. For instance, such a control mechanism can be desirable for those applications where the computerized results will be verified by a human expert and thus Ta can be set to a sufficiently low value, i.e., $Ta = 0.1$, so that the *Sensitivity* is maximized (e.g., $Sen > 0.96$) in order to accurately detect almost *all* abnormal records in the dataset. This is another crucial advantage over the competing method as it basically lacks such a control mechanism.

One performance metric, which has not also been considered in the Physionet/CinC competition is the Positive Predictivity (Precision). The range, $0.63 < Ppr < 0.75$, indicates a high level of false alarms (mis-classification of normal records). This is in fact an expected outcome for this problem considering the inter-class similarities and intra-class variations as displayed in some of the examples in Fig. 1. The low-SNR dataset is formed from the PCG records with the highest noise variance as the PCG beats of two sample records are shown in Fig. 8. It is clear that the entire morphology of a regular PCG beat is severely degraded.

Besides this insignificant time complexity for training, another crucial advantage of the proposed approach over the competing method is its elegant computational efficiency for the (online) beat classification, which is the main process of PCG monitoring. The average time for a single PCG beat classification is only 0.29 msec. Since the human heart beats 1–4 times in a second, the classification of the PCG beat(s) acquired in a real-time system will take an insignificant time which is a mere fraction of a second. For instance, for 1 beat/second, this indicates more than 3000× higher speed than a real-time requirement with the aforementioned workstation. A single-CPU execution time will then be $<39 \times 0.2$

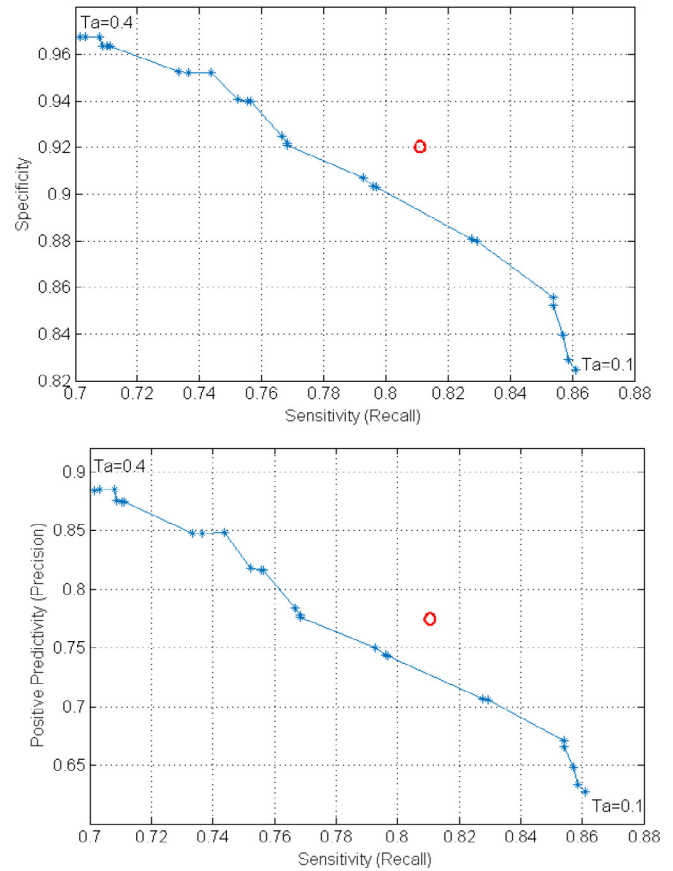


Fig. 8. 4 consecutive N beats from a sample PCG record in low-SNR dataset.

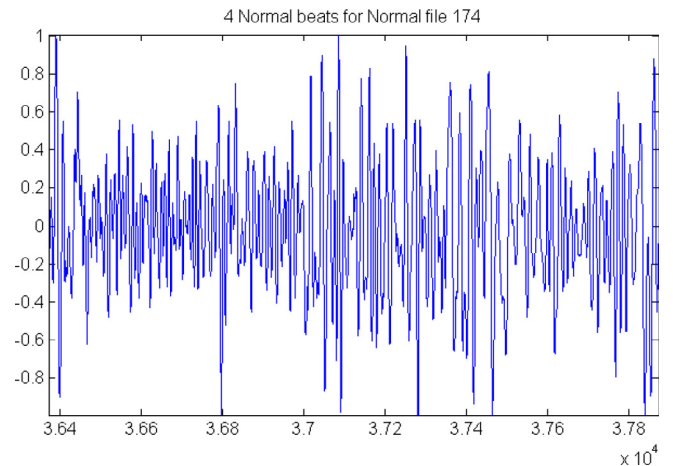


Fig. 9. Average Specificity (top) and Positive Predictivity (bottom) vs. Sensitivity (Recall) plots with varying Ta in the range [0.1, 0.4] for the test partitions of 4-fold cross validation over the low-SNR dataset. The red circle represents the performance of Zabihi et al. [40].

9 ms = 11.3 ms. Therefore, the real-time processing capability (the ability to process a beat before the next beat is acquired) for PCG monitoring can be feasible even with low-power portable computers.

6. Conclusions

Recently the *PhysioNet (CinC) Challenge* set the state-of-the-art for anomaly detection in PCG records. The best anomaly detection

method can classify each PCG record as a whole using a wide range of features over an ensemble of classifiers. In this study our aim is to set a new frontier solution which can achieve a superior detection performance. As additional but equally important objectives that have not been tackled in the challenge, the proposed systematic approach is designed to provide a real-time solution whilst reducing the false alarms.

In this study we also investigated the fundamental drawback of such global and automated solutions including the one proposed in this paper. This is a well-known fact for ECG classification that has been reported by several recent studies [14,12,17,25,23,27]. Similar to ECG signal, PCG, too, shows a wide range of variations among patients as some examples are shown in Fig. 1. In other words, PCG also exhibits patient-specific patterns many of which can conflict with other patient's data during classification. This drawback is also the main reason why further significant performance improvement may no longer be viable with such global approaches.

As a future work, similar to the state-of-the-art ECG studies, we shall address this drawback with a patient-specific or personalized PCG anomaly detection approach that can further improve this frontier in a significant way. Moreover, the focus will particularly be drawn on novel machine learning paradigms based on heterogeneous network architectures with non-linear neurons such as Operational Perceptrons [20,21,34–36] and Operational Neural Networks [21]. In this way, we can expect that the classification performance will significantly be improved whilst the network depth and complexity can further be reduced.

CRedit authorship contribution statement

Serkan Kiranyaz: Writing - review & editing, Conceptualization, Methodology, Software. **Morteza Zabih:** Writing - review & editing, Conceptualization, Methodology, Software. **Ali Bahrami Rad:** Writing - review & editing, Conceptualization, Methodology, Software. **Turker Ince:** Conceptualization, Methodology, Writing - review & editing. **Ridha Hamila:** Writing - review & editing. **Moncef Gabbouj:** Writing - review & editing, Supervision.

Acknowledgements

This work has been supported by Qatar National Research Fund (QNRF) over the ongoing project, NPRP11S-0108-180228.

References

- [1] M.S.O. Abdeljaber, O. Avci, S. Kiranyaz, B. Boashash, H. Sodano, D.J. Inman, 1-D CNNs for structural damage detection: verification on a structural health monitoring benchmark data, *Neurocomputing* (Elsevier) 75 (2018) 154–170, <https://doi.org/10.1016/j.jsv.2016.10.043>.
- [2] M.S.O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, D.J. Inman, Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks, *J. Sound Vib.* (Elsevier) 388 (2017) 154–170, <https://doi.org/10.1016/j.jsv.2016.10.043>.
- [3] O. Abdeljaber, S. Sassi, O. Avci, S. Kiranyaz, A. Ibrahim, M. Gabbouj, Fault detection and severity identification of ball bearings by online condition monitoring, *IEEE Trans. Ind. Electron.* (2018), <https://doi.org/10.1109/TIE.2018.2886789>.
- [4] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, D.J. Inman, Wireless and real-time structural damage detection: a novel decentralized method for wireless sensor networks, *J. Sound Vib.* (Elsevier) 424 (2018) 158–172, <https://doi.org/10.1016/j.jsv.2018.03.008>.
- [5] C.C. Balili, M. Sobrepena, P.C. Naval, Classification of heart sounds using discrete and continuous wavelet transform and random forests, *Asian Conf. on Pattern Recognition*. Kuala Lumpur, 2015.
- [6] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhuber, Deep big simple neural nets for handwritten digit recognition, *Neural Comput.* 22 (12) (2010) 3207–3220.
- [7] S.W. Deng, J.Q. Han, Towards heart sound classification without segmentation via autocorrelation feature and diffusion maps, *Future Gener. Comput. Syst.* 60 (2016) 13–21.
- [8] L. Eren, T. Ince, S. Kiranyaz, A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier, *J. Signal Process. Syst.* (Springer) (2018) 1–11, <https://doi.org/10.1007/s11265-018-1378-3>.
- [9] C.N. Guptaa, R. Palaniappan, S. Swaminathana, S.M. Krishnan, Neural network classification of homomorphic segmented heart sounds, *Appl. Soft Comput.* (Elsevier) 1 (2007) 286–297.
- [10] M.M. Hassan, Md.G.R. Alam, Z.Md. Uddin, S. Huda, A. Almogrena, G. Fortino, Human emotion recognition using deep belief network architecture, *Inform. Fusion* 51 (2019) 20–28.
- [11] L. Huiying, L. Sakari, H. Iiro, A heart sound segmentation algorithm using wavelet, *19th Int. Conf. IEEE/EMBS*. Chicago, IL, 1997.
- [12] T. Ince, S. Kiranyaz, M. Gabbouj, A Generic and robust system for automated patient-specific classification of electrocardiogram signals, *IEEE Trans. Biomed. Eng.* 56 (5) (2009) 1415–1426.
- [13] T. Ince, S. Kiranyaz, L. Eren, M. Askar, M. Gabbouj, Real-time motor fault detection by 1D convolutional neural networks, *IEEE Trans. Ind. Electron.* 63 (1) (2016) 7067–7075, <https://doi.org/10.1109/TIE.2016.2582729>.
- [14] W. Jiang, S.G. Kong, Block-based neural networks for personalized ECG signal classification, *IEEE Trans. Neural Networks* 18 (6) (2007) 1750–1761.
- [15] W.C. Kao, C.C. Wei, Automatic phonocardiograph signal analysis for detecting heart valve disorders, *Expert Syst. Appl.* 38 (6) (2011) 6458–6468.
- [16] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Alemadi, M. Gabbouj, Real-time fault detection and identification for MMC using 1D convolutional neural networks, *IEEE Trans. Ind. Electron.* (2018), <https://doi.org/10.1109/TIE.2018.2833045>.
- [17] S. Kiranyaz, T. Ince, M. Gabbouj, Multi-dimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition, Springer, 2013.
- [18] S. Kiranyaz, T. Ince, M. Gabbouj, Real-time patient-specific ECG classification by 1D convolutional neural networks, *IEEE Trans. Biomed. Eng.* 99 (2015), <https://doi.org/10.1109/TBME.2015.2468589>.
- [19] S. Kiranyaz, T. Ince, M. Gabbouj, Personalized monitoring and advance warning system for cardiac arrhythmias scientific reports, *Nature* 7 (2017), <https://doi.org/10.1038/s41598-017-09544-z> (SREP-16-52549-T).
- [20] S. Kiranyaz, T. Ince, A. Iosifidis, M. Gabbouj, Progressive operational perceptrons, *Neurocomputing* (Elsevier) 224 (2017) 142–154, <https://doi.org/10.1016/j.neucom.2016.10.044>.
- [21] S. Kiranyaz, T. Ince, A. Iosifidis, M. Gabbouj, Operational Neural Networks", *Neural Computing and Applications* (Springer-Nature) (Mar. 2020) 1–24, <https://doi.org/10.1007/s00521-020-04780-3>.
- [22] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems* (NIPS), 2012.
- [23] C. Li, C.X. Zheng, C.F. Tai, Jan), Detection of ECG characteristic points using wavelet transforms, *IEEE Trans. Biomed. Eng.* 42 (1) (1995) 21–28.
- [24] C. Liu, D. Springer, Q. Li, et al., An open access database for the evaluation of heart sound algorithms, *Physiol. Meas.* 2181–2213 (2016), <https://doi.org/10.1088/0967-3334/37/12/2181>.
- [25] M. Llamedo, J.P. Martinez, An automatic patient-adapted ECG heartbeat classifier allowing expert assistance, *IEEE Trans. Biomed. Eng.* 59 (8) (2012) 2312–2320.
- [26] D.J. MacKay, Bayesian interpolation, *Neural Comput.* 4 (3) (1992) 415–447.
- [27] T. Mar, S. Zaunseder, J.P. Martinez, M. Llamedo, R. Poll, Optimization of ECG classification by means of feature selection, *IEEE Trans. Biomed. Eng.* 58 (8) (2011) 2168–2177.
- [28] M. McConnell, *Pediatric Heart Sounds*, Springer Science & Business Media, 2008.
- [29] A. Moukadem, A. Dieterlen, N. Hueber, C. Brandt, A robust heart sounds segmentation module based on S-transform, *Biomed. Signal Process. Control*, Elsevier 8 (3) (2013) 273–281, <https://doi.org/10.1016/j.bspc.2012.11.008>.
- [30] Physionet. (2016, Sep 8). Retrieved from Physionet/CinC Challenge 2016: <https://physionet.org/challenge/2016/>.
- [31] F. Sattar, F. Jin, A. Moukadem, C. Brandt, A. Dieterlen, Time-Scale Based Segmentation for Degraded PCG Signals Using NMF, Non-Negative Matrix Factorization: Theory and Applications, Springer Publisher, 2015.
- [32] D. Scherer, A. Muller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, *Int. Conf. on Artificial Neural Networks*, 2010.
- [33] G. Serpen, Z. Gao, Complexity analysis of multilayer perceptron neural network embedded into a wireless sensor network, *Procedia Comput. Sci.* 36 (2014) 192–197, <https://doi.org/10.1016/j.procs.2014.09.078>.
- [34] D.T. Tran, S. Kiranyaz, M. Gabbouj, A. Iosifidis, Heterogeneous multilayer generalized operational perceptron, *IEEE Trans. Neural Networks Learn. Syst.* (2019), <https://doi.org/10.1109/TNNLS.2019.2914082> (To Appear).
- [35] D.T. Tran, S. Kiranyaz, M. Gabbouj, A. Iosifidis, Progressive operational perceptron with memory, *Neurocomputing* (Elsevier) (2019), <https://doi.org/10.1016/j.neucom.2019.10.079>.
- [36] D.T. Tran, S. Kiranyaz, M. Gabbouj, A. Iosifidis, PyGOP: a python library for generalized operational perceptron, *Knowledge-Based Systems* (Elsevier) 182 (2019), <https://doi.org/10.1016/j.knsys.2019.06.009>.
- [37] Z.Md. Uddin, M.M. Hassan, Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network, *IEEE Sens. J.* 19 (19) (2019) 8413–8419, <https://doi.org/10.1109/JSEN.2018.2871203>.
- [38] Z.Md. Uddin, M.M. Hassan, A. Alsanad, C. Savaglio, A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust, *Healthcare* 55 (2019) 105–115.

- [39] S. Yuenyong, A. Nishihara, W. Kongprawechnon, K. Tungpimolrut, A framework for automatic heart sound analysis without segmentation, *Biomed. Eng. Online* (2011) 10–13.
- [40] M. Zabihi, A.B. Rad, S. Kiranyaz, M. Gabbouj, A.K. Katsaggelos, Heart sound anomaly and quality detection using ensemble of neural networks without segmentation, in: *Computing in Cardiology Conference, CinC 2016, IEEE, Vancouver, BC, 2016*, pp. 613–616, <https://doi.org/10.23919/CIC.2016.7868817>.
- [41] D. Zhang, J. He, Y. Jiang, M. Du, Analysis and classification of heart sounds with mechanical prosthetic heart valves based on Hilbert-Huang transform, *Int. J. Cardiol.* (2011) 126–127.



Serkan Kiranyaz (SM'13) is a Professor in Qatar University, Doha, Qatar. He published 2 books, 5 book chapters, more than 70 journal papers in high impact journals, and 100 papers in international conferences. He made contributions on evolutionary optimization, machine learning, bio-signal analysis, computer vision with applications to recognition, classification, and signal processing. Prof. Kiranyaz has co-authored the papers which have nominated or received the "Best Paper Award" in ICIP 2013, ICPR 2014, ICIP 2015 and IEEE TSP 2018. He had the most-popular articles in the years 2010 and 2016, and most-cited article in 2018 in

IEEE Transactions on Biomedical Engineering. During 2010–2015 he authored the 4th most-cited article of the *Neural Networks* journal. His research team has won the 2nd and 1st places in PhysioNet Grand Challenges 2016 and 2017, among 48 and 75 international teams, respectively. In 2019, he won the "Research Excellence Award" and "Merit Award" of Qatar University. His theoretical contributions to advance the current state of the art in modelling and representation, targeting high long-term impact, while algorithmic, system level design and implementation issues target medium and long-term challenges for the next five to ten years. He in particular aims at investigating scientific questions and inventing cutting edge solutions in "personalized biomedicine" which is in one of the most dynamic areas where science combines with technology to produce efficient signal and information processing systems meeting the high expectation of the users.



Morteza Zabihi was born in Iran in 1988. He received his M.Sc. degree in biomedical engineering from Tampere University, Tampere, Finland, in 2013. He is currently graduating from his Ph.D. and working as a researcher in the Department of Computing Sciences at Tampere University, Tampere, Finland. His research interests include nonlinear dynamics and time series analysis, pattern recognition, and machine learning with application to EEG and ECG signal processing. Since 2015, he has been ranked in the top three teams in four international competitions, such as PhysioNet/Computing in Cardiology and IEEE EMBS Neural Engineering Brain-Computer Interface Challenges.



Ali Bahrami Rad received the B.Sc. degree in Electrical Engineering from Imam Hossein University, Tehran, Iran in 2003, the M.Sc. degree in Biomedical Engineering from Tampere University of Technology, Tampere, Finland in 2011, and the PhD degree in Computer Engineering-Information Technology from the University of Stavanger, Stavanger, Norway in 2017. He completed his postdoctoral research in the Department of Electrical Engineering & Automation at Aalto University, Espoo, Finland. He is currently a Visiting Assistant Professor in the Department of Biomedical Informatics at Emory University, Atlanta, USA. His current research

interests include machine learning, neural systems, and biomedical/electrophysiological signal analysis with applications to resuscitation, defibrillation, seizure

detection and stem-cell technology. He is the recipient of multiple international prizes in biosignal analysis and machine learning contests including PhysioNet/Computing in Cardiology Challenge 2017 (tied for the 1st place), PhysioNet/Computing in Cardiology Challenge 2016 (2nd place), and Brain-Computer Interface Challenge at IEEE EMBS Neural Engineering Conference 2015 (3rd place).



Turker Ince received the B.S. degree from the Bilkent University, Ankara, Turkey, in 1994, the M.S. degree from the Middle East Technical University, Ankara, Turkey, in 1996, and the Ph.D. degree from the University of Massachusetts, Amherst (UMass-Amherst), in 2001 all in electrical engineering. From 1996 to 2001, he was a Research Assistant at the Microwave Remote Sensing Laboratory, UMass-Amherst. He worked as a Design Engineer at Aware, Inc., Boston, from 2001 to 2004, and at Texas Instruments, Inc., Dallas, from 2004 to 2006. In 2006, he joined the faculty of Engineering and Computer Science at Izmir University of Economics, Turkey, where he is currently an Associate Professor. His research interests include radar remote sensing and target recognition, signal processing, evolutionary optimization and machine learning.



Prof. Ridha Hamila (QU received the M.Sc., Licentiate of Technology with distinction, and Doctor of Technology degrees from Tampere University of Technology (TUT), Tampere, Finland, in 1996, 1999, and 2002, respectively. Dr. Ridha Hamila is currently a Full Professor at the Department of Electrical Engineering, Qatar University, Qatar. From 1994 to 2002 he held various research and teaching positions at TUT within the Department of Information Technology, Finland. From 2002 to 2003 he was a System Specialist at Nokia research Center and Nokia Networks, Helsinki. From 2004 to 2009 he was with Emirates Telecommunications Corporation, UAE.

Also, from 2004 to 2013 he was adjunct Professor at the Department of Communications Engineering, TUT. His current research interests include mobile and broadband wireless communication systems, Internet of Everything, and Machine Learning. In these areas, he has published over 150 journal and conference papers most of them in the peer reviewed IEEE publications, filed five US patents, and wrote numerous confidential industrial research reports. Dr. Hamila has been involved in several past and current industrial projects, Ooreedo, Qatar National Research Fund, Finnish Academy projects, EU research and education programs. He supervised a large number of under/graduate students and postdoctoral fellows. He organized many international workshops and conferences. He is a Senior Member of IEEE. Prof. Ridha is in close collaboration with the LPI and other international team members. He will be responsible of designing reliable and secure communication protocols between the client and server of the proposed system.



Moncef Gabbouj (F'11) is a well-established world expert in the field of image processing, and held the prestigious post of Academy of Finland Professor during 2011–2015. He has been leading the Multimedia Research Group for nearly 25 years and managed successfully a large number of projects in excess of 18M Euro. He has supervised 45 PhD theses and over 50 MSc theses. He is the author of several books and over 700 papers. His research interests include Big Data analytics, multimedia content-based analysis, indexing and retrieval, artificial intelligence, machine learning, pattern recognition, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding. Dr. Gabbouj is a Fellow of the IEEE and member of the Academia Europaea and the Finnish Academy of Science and Letters. He is the past Chairman of the IEEE CAS TC on DSP and committee member of the IEEE Fourier Award for Signal Processing. He served as associate editor and guest editor of many IEEE, and international journals.