

# Automated Atrial Fibrillation Detection using a Hybrid CNN-LSTM Network on Imbalanced ECG Datasets

Georgios Petmezas<sup>a</sup>, Kostas Haris<sup>a</sup>, Leandros Stefanopoulos<sup>a</sup>, Vassilis Kilintzis<sup>a</sup>,  
Andreas Tzavelis<sup>b,c</sup>, John A Rogers<sup>b,c</sup>, Aggelos K Katsaggelos<sup>b</sup>, Nicos Maglaveras<sup>a,b,\*</sup>

<sup>a</sup> Lab of Computing, Medical Informatics and Biomedical Imaging Technologies Aristotle University of Thessaloniki, Thessaloniki, Greece

<sup>b</sup> Dept of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA

<sup>c</sup> Dept of Material Science, Northwestern University, Evanston, IL, USA

## ARTICLE INFO

### Keywords:

atrial fibrillation  
arrhythmia detection  
CNN  
LSTM  
focal loss

## ABSTRACT

Atrial fibrillation is a heart arrhythmia strongly associated with other heart-related complications that can increase the risk of strokes and heart failure. Manual electrocardiogram (ECG) interpretation for its diagnosis is tedious, time-consuming, requires high expertise, and suffers from inter- and intra-observer variability. Deep learning techniques could be exploited in order for robust arrhythmia detection models to be designed. In this paper, we propose a novel hybrid neural model utilizing focal loss, an improved version of cross-entropy loss, to deal with training data imbalance. ECG features initially extracted via a Convolutional Neural Network (CNN) are input to a Long Short-Term Memory (LSTM) model for temporal dynamics memorization and thus, more accurate classification into the four ECG rhythm types, namely normal (N), atrial fibrillation (AFIB), atrial flutter (AFL) and AV junctional rhythm (J). The model was trained on the MIT-BIH Atrial Fibrillation Database and achieved a sensitivity of 97.87%, and specificity of 99.29% using a ten-fold cross-validation strategy. The proposed model can aid clinicians to detect common atrial fibrillation in real-time on routine screening ECG.

## 1. Introduction

According to the World Health Organization (WHO), cardiovascular diseases are the number one worldwide cause of death, enumerating around 17.9 million victims on a yearly basis [1]. Atrial fibrillation (AFib) is the most common arrhythmia diagnosed in clinical practice [2]. It is characterized by a rapid, irregular and heterogeneous electrical activity of the heart that is caused by ineffective atrial contractions [3]. In the general population, AFib is closely related to increased mortality and can trigger ischaemic strokes, secondary thromboembolic events, sometimes even heart failure [4]. It mainly affects male patients and its prevalence increases analogously to age [5]. In fact, it is estimated that due to overaging the total number of AFib patients will double by the year 2050 [6].

Electrocardiography is the most frequently used tool for diagnosing AFib [7]. Electrocardiogram (ECG) is a digital signal recorded by placing electrodes on the surface of the human body in order that voltage changes caused by the electrical activity of the heart can be detected [8]. Traditional diagnosis usually employs ECG interpretation, which makes

a demand of using very accurate devices and high expertise from the doctors' perspective [9]. Conventionally, cardiologists visually inspect the 12-lead ECGs waveforms in a digital image format. Nonetheless, it is quite often that ECG signals lasting many hours even days need to be analysed. As it can be easily noted, this is a very time-consuming and wearing procedure that significantly limits the diagnosis' impartiality. This limitation can be lifted by using computational techniques for automated arrhythmia detection and ECG classification.

In the last decades there has been a significant growth of interest in the field of automated classification of various heart arrhythmias based on ECG signals proposing machine learning (ML) techniques including support vector machines (SVM) [10], k-nearest neighbors algorithm (KNN) [11], principal component analysis (PCA) [12] and adaptive backpropagation neural network among others [13]. These techniques though had a number of constraints including computational limitations in the learning process which would restrict their adaptation capacity in personalized health systems.

Convolutional neural networks (CNN), which perform both feature extraction and classification, are widely applied in studies that

\* Corresponding author at: Lab of Computing, Medical Informatics and Biomedical Imaging Technologies Aristotle University of Thessaloniki, Thessaloniki, Greece.  
E-mail address: [nicmag@med.auth.gr](mailto:nicmag@med.auth.gr) (N. Maglaveras).

implement deep learning (DL) techniques [14]. Regarding the structure of a typical CNN, numerous convolutional and pooling layers are placed in succession forming a deep network that is capable of extracting the underlying features of a single input and reducing its dimensionality at the same time. These render them ideal for studies with huge computational demands such as ECG classification [15,16].

Besides CNN, Long Short-Term Memory (LSTM) networks [17] are another type of DL techniques frequently used for time series analysis [18], including speech recognition [19] and language processing [20]. LSTM networks are a subcategory of the recursive neural networks (RNN) and constitute an evolution of the classic RNN. The advantage of using LSTM networks against other types of neural networks (NN) is twofold – they can not only learn the temporal dynamics of the input data, but also can selectively remember or forget information checking the current memory state – and that made them popular for ECG signal classification [18,21].

The purpose of the present study is to propose a hybrid DL model architecture that combines CNN and LSTM in order to classify different types of cardiovascular arrhythmias. The main idea behind this approach is to use the CNN part as feature extractor that supplies the LSTM part with the most discriminative features of the input achieving in parallel dimensionality reduction. In addition, the proposed model utilizes the focal loss (FL) function to achieve both prediction error reduction and data imbalance handling. Training and evaluation of the model are performed on ECG signals originating from the well-known MIT-BIH Atrial Fibrillation Database [22,23] and besides normal beats, three types of arrhythmias are detected, namely atrial fibrillation (AFIB), atrial flutter (AFL) and junctional rhythm (J).

The remainder of this paper is structured as follows: In Section 2, background knowledge for atrial fibrillation is presented and related state-of-the-art deep learning techniques for ECG classification are reviewed. In Section 3, the research methodology is described in detail, in Section 4 the performance evaluation procedure of the proposed model is presented and, in Section 5, its performance is discussed and numerically compared to relevant recent studies. Finally, in Section 6 conclusions of the study are summarized.

## 2. Literature review

### 2.1. Atrial fibrillation

Atrial fibrillation (AFib) is a cardiovascular disease in which the atria beat out of control and coordination with the ventricles [24]. It is the most common arrhythmia in clinical practice, which justifies the fact that many countries, especially these of the western world, allocate high budgets for handling it [25]. In fact, recent studies have predicted that AFib will affect 6-12 million of people in the USA by the year 2050 and around 18 million in Europe by the year 2060 [26–28].

AFib is closely associated with other heart complications and can significantly increase the risk of strokes, transient ischemic attack (TIA) and heart failure [29,30]. Risk factors that increase its prevalence include hypertension, coronary artery disease, cardiomyopathy, diabetes, obesity, hyperthyroidism, excessive alcohol consumption and the personal history of other cardiovascular diseases. Also, age, family medical history, reduced body mass and height can as well contribute to the development of AFib [31].

There are several types of AFib depending on the extent to which it affects each patient. More precisely, there is paroxysmal AFib, a temporal episode of AFib that lasts less than a week and usually does not require treatment, persistent AFib that lasts longer than a week or less, if proper medication is taken, long-standing persistent AFib, where the heart beats abnormally for longer than a year, and, finally, permanent AFib [32].

### 2.2. Automated arrhythmia detection

Cardiovascular arrhythmias detection based on ECG signals is apparently a critical issue. As of now, a considerable number of studies has been carried out regarding the automated classification of heart arrhythmias by using wavelets [33], filter banks [34], support vector machines (SVM) [35], and other ML techniques [36,37].

ML techniques facilitate scientists' efforts by significantly reducing computational time needed to train a machine (e.g. personal computer) in order that it can learn data and make predictions on new ones by using three types of learning, namely supervised, unsupervised and reinforcement [38]. However, traditional ML techniques require manual feature extraction, which is a challenging task, especially for scientists with low expertise. On the other hand, models that implement DL methods are considered a more suitable approach for ECG classification, especially when large datasets are available.

DL techniques can overcome the problem of feature extraction, as they do not make a demand on selecting specific features manually, but automatically extract the most significant features from their input [39]. A DL model, usually, consists of a body of processing layers that can interpret data features via multiple stages of subtraction [40]. Hence, during the last few years, many scientists have concentrated their efforts on developing robust and highly accurate neural network models for the classification of arrhythmias, such as CNN or LSTM.

### 2.3. Convolutional neural networks

Convolutional neural networks (CNN) are deep learning algorithms that are highly inspired by the way that biological neural systems (e.g. human brain) work. CNN are capable of learning step by step the spatial hierarchy of the data by memorizing high- and low-level patterns.

Typically, a CNN is a mathematical structure that consists of three types of layers that are lined up: convolutional, pooling and fully-connected layers. Convolutional and pooling layers perform feature extraction and dimensionality reduction, respectively, while fully-connected layers map the extracted features to predict a final output (e.g. the class that the input belongs to).

Regarding the CNN architecture, each layer feeds its output into the next one. Layer outputs become more complex as the information passes deeper into the network. This process is called training and its purpose is to minimize the difference between network predictions and ground truth labels via an optimization algorithm that combines back propagation and gradient descent among others.

During the last few years, CNN are widely implemented to deal with ECG classification and arrhythmia detection. More precisely, Li et al. [41] designed a six-layered 1D CNN to classify different types of cardiac arrhythmias consisting of an input layer, two convolutional layers, two down-sampling layers, one fully-connected layer and an output layer, using the MIT-BIH Arrhythmia Database [42,43]. Acharya et al. [45] made use of the same database in order to achieve beat classification by a CNN containing three convolutional, three max-pooling and three fully-connected layers. Ochiai et al. [44] developed a seven-layered 1D CNN model to classify the same ECG signals using three sets of a convolutional and a pooling layer, one flatten layer and two fully-connected layers. Furthermore, Savalia and Emamian (2018) trained a six-layered CNN using the MIT-BIH Normal Sinus Rhythm Database [23] and the MIT-BIH Arrhythmia Database in order to separate normal and abnormal heartbeats, while Pourbabaee et al. [46] suggested a five-layered CNN to detect paroxysmal AFib using the PAF Prediction Challenge Database [47].

On the other hand, several authors made an attempt to classify ECG signals by transforming ECG beats into 2D images. Thus, Jun et al. [48] designed an eleven-layered 2D CNN for ECG image classification by using three sets of two convolutional and one pooling layers, one dense and one output layer, while Xu et al. [49] implemented Modified Frequency Slice Wavelet Transform (MFSWT) to construct images out of

1-sec ECGs from the MIT-BIH Atrial Fibrillation Database [22] and fed them into a twelve-layered 2D CNN.

#### 2.4. Long short-term memory networks

Long short-term memory networks (LSTM) are a subcategory of recursive neural networks (RNN) that are capable of learning the long-term dependencies between data, which makes them perfect for cases where information must be maintained for a long time in order for accurate predictions to be made. a problem that classic RNN cannot overcome easily.

LSTM can also solve the vanishing gradient problem [50], that is to say the saturation of past layer neurons as the information passes deeper into the network. This means that neurons that hold valuable information, but lie in remote time points, stop contributing to the network resulting in inefficient learning and, finally, performance saturation.

Regarding the LSTM structure, each network consists of memory blocks, also called cells. Each cell transfer two states to the next one, cell and hidden state. Notwithstanding that the cell state enables unalterable forward data-flow, data can be modified by several types of transformation. More precisely, data can be added or removed from the cell state via sigmoid gates. A gate acts as a layer of neurons that contain different weights. In this way, LSTM can solve the problem of long-term dependencies, as they implement gates to control the process of memorization.

Conventionally, ECG is a time series strongly characterized by long-term dependencies between its values at different time points. Many scientists have tried to make the most of it by designing LSTM models in order to classify cardiovascular arrhythmias based on these dependencies. More in detail, Gao et al. [51] implemented a network consisting of a LSTM layer and two fully-connected layers to classify beats from the MIT-BIH Arrhythmia Database into eight categories. Yildirim [52] fed the same database into two bidirectional LSTM and two fully-connected layers to detect five types of ECG beats, while Kim and Pyun [53] tested the performance of six different LSTM network architectures in order to achieve ECG classification for the MIT-BIH Normal Sinus Rhythm and the MIT-BIH Arrhythmia Databases.

#### 2.5. Data augmentation

In general, data imbalance constitutes a major problem during the training process as it can cause overfitting. This means that the model fits perfectly to the training data, but still cannot generalize to new data. During the last few years, there has been a great deal of interest in solving this problem, mainly by using several oversampling techniques.

The most known oversampling technique is random oversampling and undersampling (ROU). Nonetheless, the main disadvantage with this approach is that it can by itself lead to overfitting for high imbalanced datasets. Synthetic Minority Over-sampling Technique (SMOTE) [54] is another widely used oversampling technique. It is based on the calculation of the nearest neighbors by implementing the Euclidean distance between data and is capable of creating synthetic data by multiplying nearest neighbors by a vector containing values between 0 and 1. Last but not least, recently proposed oversampling techniques include generative oversampling method (GenOMe) [55], that creates new data based on known distributions (beta, gamma, gaussian), Synthetic Minority Oversampling Technique with random undersampling (SMOTE + RU) and distribution-based balancing (DBB) [56].

### 3. Methodology

#### 3.1. MIT-BIH Atrial Fibrillation Database

In the present study, ECG signals from the MIT-BIH Atrial Fibrillation have been used. This database includes 23 long-term 2-lead ECG recordings of subjects with AFib (mostly paroxysmal) with a sampling

frequency of 250 Hz. The data collection was carried out at the Boston's Beth Israel Hospital. Expert cardiologists prepared separate rhythm annotation files in which they marked 4 types of ECG rhythms, namely normal (N), atrial fibrillation (AFIB), atrial flutter (AFL) and AV junctional rhythm (J) as shown in Table 1.

#### 3.2. Data preprocessing

The first step of data preprocessing usually involve a noise removal technique. In general, there are two major types of noise that can corrupt an ECG signal, high-frequency, such as powerline interference and Gaussian white noise, and low-frequency noise, including baseline wander noise among others.

During the last decades, several noise removal techniques have been proposed. More specifically, FIR and IIR filters are widely used for signal denoising, mainly because of their capability of cutting off specific frequency bands where noise is present [57,58]. They are uniquely characterized by their frequency response  $H(\omega)$  in the frequency domain, that is to say the discrete Fourier transform (DFT) of their time response  $h(t)$  in the time domain [59].

Butterworth filters are one of the most common types of digital filters [60]. They were firstly proposed in 1930 and are widely known for their flat frequency response. The magnitude-squared function of a normalized lowpass Butterworth filter with a cutoff frequency  $\omega_c = 1$  rad/sec is defined as follows:

$$|H(\omega)|^2 = \frac{1}{1 + \omega^{2n}} \quad (3.2.1)$$

where  $n$  is the filter's order. Given the magnitude-squared function, the transfer function of a Butterworth filter can be found as follows:

$$H(s)H(-s) = \frac{1}{1 + (-s^2)^n} \quad (3.2.2)$$

Moreover, wavelet transform (WT) has multiple applications in non-static signal processing [61,62] and that made it a very popular tool for ECG processing [63]. WT comprises a generalization of the short-time Fourier transform (STFT). It could be described as the process of finding the projection of a signal on a set of base functions, known as wavelets. However, in contrast to STFT which analyzes the signal using square windows in the time-frequency domain, WT provides high frequency resolution for the low frequencies and high temporal resolution for the high frequencies of the signal.

Discrete wavelet transform (DWT) of a signal  $x(n)$  is defined based on the approximation coefficients,  $W_\varphi(j_0, k)$ , and the detail coefficients,  $W_\psi(j, k)$ , as follows:

$$W_\varphi(j_0, k) = \frac{1}{\sqrt{M}} \sum_n x(n) \varphi_{j_0, k}(n) \quad (3.2.3)$$

$$W_\psi(j, k) = \frac{1}{\sqrt{M}} \sum_n x(n) \psi_{j, k}(n) \text{ for } j \geq j_0 \quad (3.2.4)$$

while the inverse discrete wavelet transform (IDWT) is given by:

$$x(n) = \frac{1}{\sqrt{M}} \sum_k W_\varphi(j_0, k) \varphi_{j_0, k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^J \sum_k W_\psi(j, k) \psi_{j, k}(n) \quad (3.2.5)$$

**Table 1**

The MIT-BIH Atrial Fibrillation Database.

Label	class	number of beats
0	N	619345
1	AFIB	345241
2	AFL	5241
3	J	182
Total		970009

where  $n = 0, 1, 2, \dots, M-1$ ,  $j = 0, 1, 2, \dots, J-1$ ,  $k = 0, 1, 2, \dots, 2^j-1$  and  $M$  the number of samples that are about to be transformed. This number is defined as  $M = 2^J$ , where  $J$  declares the number of the transform levels. The base functions  $\varphi_{j,k}(n)$  and  $\psi_{j,k}(n)$  are given by:

$$\varphi_{j,k}(n) = 2^{j/2} \varphi(2^j n - k) \quad (3.2.6)$$

$$\psi_{j,k}(n) = 2^{j/2} \psi(2^j n - k) \quad (3.2.7)$$

where  $\varphi(n)$  is the scaling function and  $\psi(n)$  the wavelet function. In essence, DWT is a hierarchical algorithm that uses highpass and lowpass filters and signal downsampling at the same time, in order to decompose the signal in different scales. In this way, the signal is divided into two parts at each transform level, one containing the lower half of the frequency band and the other the upper half. The output of the lowpass filter corresponds to the approximation coefficients, while the output of the highpass filter represents the detail coefficients. Fig. 1 presents the ECG preprocessing method that has been applied on the ECG signals so that clean beat signals are extracted and fed to the deep learning model.

In this study, a 7<sup>th</sup> order Butterworth highpass filter with a cutoff frequency of 0.5 Hz was implemented to remove baseline wander noise and a 6<sup>th</sup> order Butterworth lowpass filter was applied to remove powerline interference and other high frequency components of the noise. Regarding Gaussian white noise, a 3-level DWT using db4 wavelet of the Daubechies family and soft thresholding was implemented. Fig. 2a presents the result of applying the denoising method on the ECG data.

To detect the R peaks of the ECG signal, a 4-level DWT was applied. More specifically, the sum of level 3 and 4 detail coefficients can efficiently locate the area where each QRS complex is present. Therefore, by applying a peak detection algorithm on the sum of D3 and D4 coefficients, the R peaks can be accurately detected, as shown in Fig. 2b.

### 3.3. Convolutional neural networks

In general, neural networks are consisted of layers of neurons that use weights. Each neuron receives a body of inputs and calculates a scalar product followed by a non-linear operation. This process is repeated for every layer up to the output layer, where the network makes prediction for the input data.

CNN are a special type of deep learning algorithms that are widely used in research studies. They are named after the convolution, which is the mathematical operation they apply on their input data. The convolution between two time-series is a linear operation that is given by:

$$y(t) = x(t) * w(t) = \int_{-\infty}^{\infty} x(\tau) w(t - \tau) d\tau \quad (3.3.1)$$

In the CNN terminology, most of the times, the first argument ( $x$ ) represents the input, while the second one ( $w$ ) the kernel or filter. The result ( $y$ ) is commonly known as feature map. The convolutional layer constitutes the main structural element of the CNN, as it is the one that carries out the most complex network operations. It includes a set of filters and, as already noted, is responsible for producing feature maps by calculating the convolution of its input with these filters.

A filter is usually a two-dimensional grid of discrete numbers that is capable of recognizing specific patterns in the input data. In essence, the convolutional layer performs matrix multiplications between the layer's input and the selected filters by shifting the filters on the input data matrix. In particular, firstly the filter is appropriately adjusted to the upper left edge of the input matrix, so that the element by element multiplication between the two matrices is feasible. The result of this operation corresponds to the first element of a new matrix that represents the layer's output, and its value defines whether the desired pattern is present or not.

The above procedure is repeated for all the input matrix elements with right shift of the filter till the input's feature map is extracted. Besides convolutional layers, the CNN architecture also includes pooling and fully-connected layers. The pooling layers are usually placed right after each convolutional layer in order to simplify the collected information. They are responsible for reducing the feature map size by creating a condensed version of the initial map that holds only the most significant information. On the other hand, the fully-connected layers are conventionally placed at the end of the network and they are in charge of categorizing the input data based on the features that are extracted from the sequence of convolutional and pooling layers.

Finally, it is worth mentioning that, as often as not, also other types of layers are implemented, such as dropout and normalization layers. The dropout layers disable some of the network's neuron in order to minimize the classification error and the network overfitting, while the normalization layers use batches to normalize the input data to accelerate the training process.

### 3.4. Long short-term memory networks

Recursive neural networks (RNN) are suitably designed networks that analyse flows of data by deploying hidden units for memorizing information regarding previous inputs. However, their main disadvantage is their failure to preserve information for long time periods.

Long short-term memory networks (LSTM) are the most common RNN variation and are capable of recognizing and retaining the long-term dependencies between the data of an input time-series for long time periods as well as the non-linear dynamics of the data. A LSTM network is composed of cells or blocks that are placed in order.

Each cell transfers two states to the next cell, the cell state and the hidden state, and includes three kinds of gates, the forget, the input and the output gate, respectively. By using these gates, the network can handle the process of adding or removing information from its cells. The following equations describe the inner functioning of a LSTM cell:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3.4.1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3.4.2)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3.4.3)$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (3.4.4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.4.5)$$

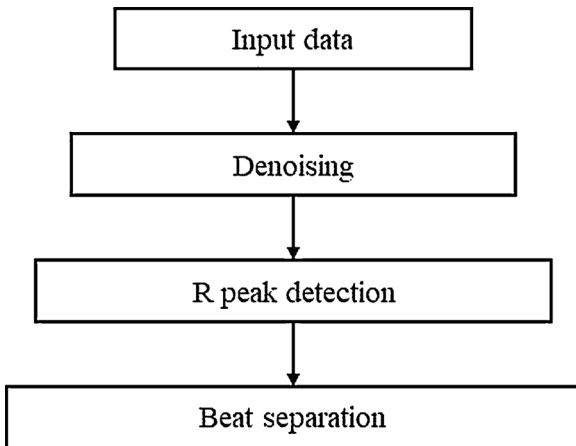


Fig. 1. Block diagram of the data preprocessing technique: at first ECG signal are being denoised by using digital filters and DWT, next a R peak detection technique that implements DWT is applied and, finally, ECG beats are extracted using 250 ms before and 500 ms after each R peak.



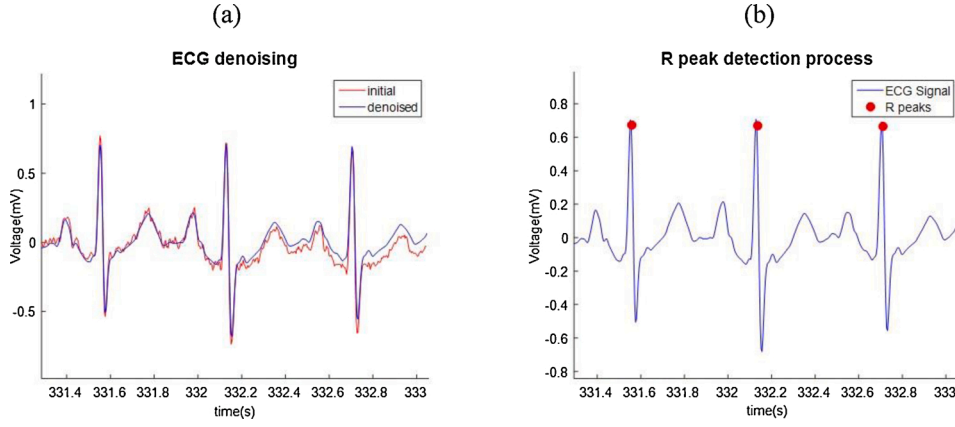


Fig. 2. ECG denoising (a) and R peak detection (b) processes.

$$h_t = o_t \tanh(C_t) \quad (3.4.6)$$

More specifically, the forget gate is in charge of removing redundant information from the cell state. It receives two inputs, the previous cell's hidden state ( $h_{t-1}$ ) and the current cell's input ( $x_t$ ), then multiplies their values with a weight matrix ( $W_f$ ) and adds a bias factor ( $b_f$ ) to this result, as given by (3.4.1). Next, a sigmoid function, that produces a vector of values between 0 and 1, is applied to the output ( $f_t$ ) in order to assign these values to the cell state.

In essence, this function decides which values of the cell state are about to be retained and which to be forgotten. In this way, if 0 is assigned to the cell state, the gate asks the network to completely forget the cell state. Equally, 1 means that the gate makes a request of completely memorizing the cell state. This procedure is repeated for the entire output,  $f_t$ , which is then multiplied to the cell state.

The input gate is responsible for adding new information to the cell state. It receives the same two arguments,  $h_{t-1}$  and  $x_t$ . The process of adding new information to the cell state is conducted in two stages.

At the first stage, the two arguments are multiplied with a new weight matrix,  $W_i$ , and a new bias factor,  $b_i$ , is added to this result. Subsequently, a new sigmoid function is applied in order that the input gate's output,  $i_t$ , is calculated, (3.4.2), and a decision is made regarding the values that should be added to cell state or not.

At the second stage, the arguments are multiplied with another weight matrix,  $W_c$ , and a new bias factor,  $b_c$ , is added to this result. By applying a tanh function, which gives values between -1 and 1, to the result of the above operations, a new vector,  $\tilde{C}_t$ , is formed, as given by (3.4.3).  $\tilde{C}_t$  includes all the new candidate values that could be added to the cell state.

Next, the result of the multiplication between these two vectors,  $i_t$  and  $\tilde{C}_t$ , is added to the vector that resulted from the multiplication between the forget gate's output and the cell state, as defined by (3.4.4). In this way, the procedure of adding new information to the cell state is concluded and the cell state is updated.

The output gate is in charge of selecting the information that will be passed as the cell's output. Much as the addition of new information to the cell state, this process can also be divided into two stages.

At the first stage, the two input arguments,  $h_{t-1}$  and  $x_t$ , are multiplied with a new weight matrix,  $W_o$ , and a new bias factor,  $b_o$ , is added, as given by (3.4.5). Hereupon, this result is multiplied with a new sigmoid function and the output gate's output,  $o_t$ , is calculated. This vector defines which values will be passed as the output to the next cell.

At the second stage, a tanh function is applied to the cell state. Finally, these two vectors are multiplied in order that the cell's output is calculated by (3.4.6).

### 3.5. Loss function

Loss function is one of the most important components of a neural network. It is used to measure the error between the model's predictions and the desired outputs. The main objective of implementing a loss function is to try minimizing this error, so that the model can accurately predict new input data. The most common loss functions are mean squared error (MSE), that is regularly used for regression models, and cross-entropy (CE) loss, which is applied to classification tasks.

In the present study, due to the high imbalance between the data, the focal loss function was implemented [64]. Focal loss (FL) is an improved version of the CE loss given by the following equation:

$$FL = - \sum_{i=1}^M y_i (1 - \hat{y}_i)^\gamma \log(\hat{y}_i), \gamma \geq 0 \quad (3.5.1)$$

where  $\gamma$  is the focusing parameter and  $(1 - \hat{y}_i)^\gamma$  the modulating factor. The advantage of using FL is twofold: it can accurately measure the classification error which is necessary when training deep learning models, and efficiently deal with data imbalance. This is because FL reduces the weights that correspond to the bigger classes, which are easily detected, so that their total contribution to the loss value is small. In this way, the model focuses on the smaller classes that traditionally are hard classified.

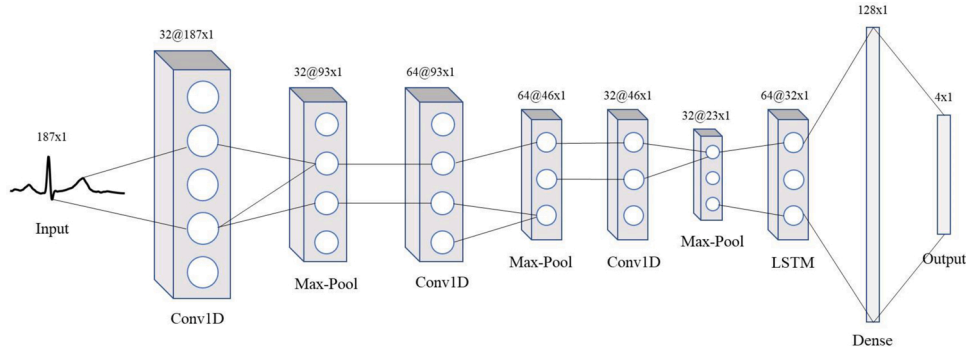
### 3.6. The proposed model

In the present study, a hybrid CNN-LSTM network that uses FL function to deal with data imbalance is proposed. The network receives ECG beats as inputs and classifies them into four distinct classes, namely N, AFIB, AFL and J.

More specifically, each ECG signal is divided into separate beats using the following procedure: firstly, all R peaks are detected and, then, each beat is defined as the 250 ms before the R peak and the 500 ms after the R peak.

Regarding the model's architecture, an input layer, thirteen hidden layers and one output layer, that predicts classes, are placed in succession, as shown in Fig. 3. In this way, the input layer receives one-dimensional ECG signals with a length of 187 samples. Firstly, the signal is passed through the CNN part of the network that is composed of triplets of layers. Each triplet includes one convolutional, one batch normalization and one max-pooling layer.

Three triplets with the above architecture are placed in order so that feature extraction is made possible. At the same time, dimensionality reduction takes place as the input passes deeper through the network. Next, the result is fed into the LSTM part of the network, that is in charge of recognizing and memorizing the long-term dependencies between the data.



**Fig. 3.** The architecture of the proposed model: the input signal is transformed by using combinations of convolutional and max-pooling layers to extract its main features, then the output is passed through a LSTM layer that memorizes the long-term dependencies between the extracted data and, finally, the output layer makes predictions for the class to which its input belongs to.

The LSTM part is composed of one LSTM layer, one flatten layer that creates one-dimensional output from multidimensional input, one fully-connected layer, one dropout layer, and, finally, one output layer that predicts one of the four classes for each input ECG beat.

The non-linear function ReLU is implemented as the activation function used in the convolutional and the fully-connected layers. Regarding the output layer, softmax activation function is applied. Softmax is a generalization of the logistic function that is widely used in the output layer of multiclass classification networks: for binary classification problems sigmoid function is implemented. The ReLU and softmax activation functions are given by the following equations:

$$R(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.6.1)$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.6.2)$$

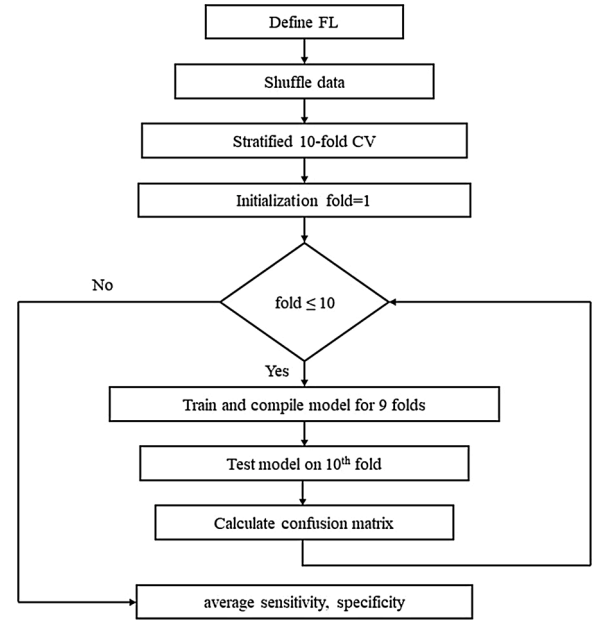
where  $z$  is a vector containing  $K$  real numbers. The model was evaluated using 10-fold cross validation. The data were fed into the network using a batch size of 128, while the learning rate was set in 0.001. Fig. 4 presents the proposed method regarding the model's training and evaluation phase using 10-fold cross validation.

#### 4. Results and Discussion

The design and implementation of the deep learning model for the automated detection of AFib were held in a Python 3.6 environment using the deep learning tool Keras. Tensorflow 2.0 was used as the backend of the Keras library. All experiments were performed on a desktop computer featuring an Intel Core i5-9600 K 3.70 GHz CPU, a 16 GB RAM memory and a 8GB NVIDIA GeForce RTX 2070 GPU.

Regarding the evaluation of the model, stratified 10-fold cross validation was employed. This means that the dataset was split into ten groups ensuring that each group has the same proportion of observations with a given categorical value. For each of the ten folds, the model was trained using the nine folds as training data and the resulting model was validated on the remaining part of the data (the 10<sup>th</sup> fold). The training process was concluded in 100 epochs and the best weights were restored for each training phase. The average epoch duration was 42 s while the model's performance evaluation on unseen data (10<sup>th</sup> fold) lasted 2 s. Fig. 5 presents the curves of the overall loss and accuracy during the training of the model using all data. As it can be easily noted, the network converges for both values after 100 epochs.

Furthermore, the model's confusion matrix was calculated. A confusion matrix, also known as an error matrix, is a C array with  $c_{ij}$  elements, where  $c_{ij}$  is the number of samples that belong to the  $i$  class and have been classified to the  $j$  class. The main diagonal of the matrix



**Fig. 4.** Block diagram of the proposed method including pseudocode: at first FL function is defined and data are shuffled, then stratified 10-fold CV is being set and the training process is repeated for all groups of folds. In each loop the model is trained for the first 9 folds and then tested on the 10<sup>th</sup> fold. Next, the confusion matrix is calculated and the values of the sensitivity and specificity are extracted. As soon as training is finished for all folds, average sensitivity and specificity are calculated.

represents the true positive predictions of the model. The overall confusion matrix for the 10-fold cross validation is presented in Fig. 6.

The confusion matrix establishes that most of the data are correctly classified, except for the J class, which includes an extremely small percentage of the overall data, though. Moreover, the network has a slight difficulty discriminating some of the data belonging to the N and the AFIB class.

Nonetheless, the confusion matrix is not capable of quantifying the model's performance, thus the calculation of some well-known evaluation metrics, as follows, is required:

$$precision = \frac{TP}{TP + FP} \quad (4.1)$$

$$accuracy = recall = sensitivity = \frac{TP}{TP + FN} \quad (4.2)$$

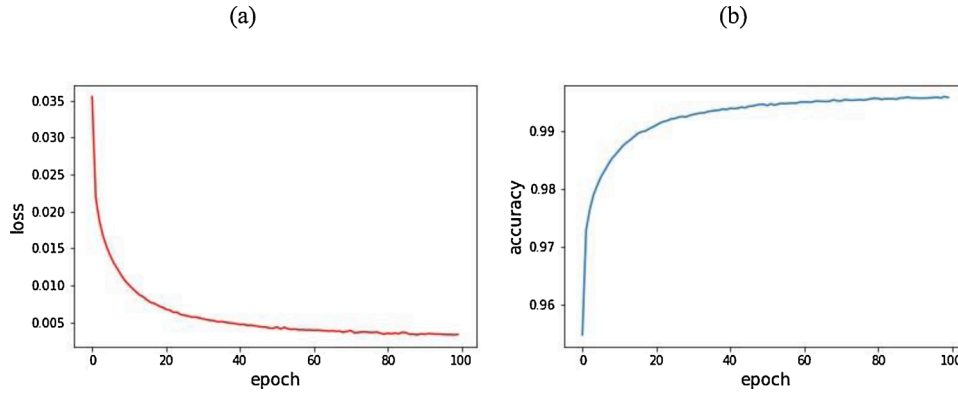


Fig. 5. Training of the CNN-LSTM model with FL: loss (a) and accuracy (b) curves.

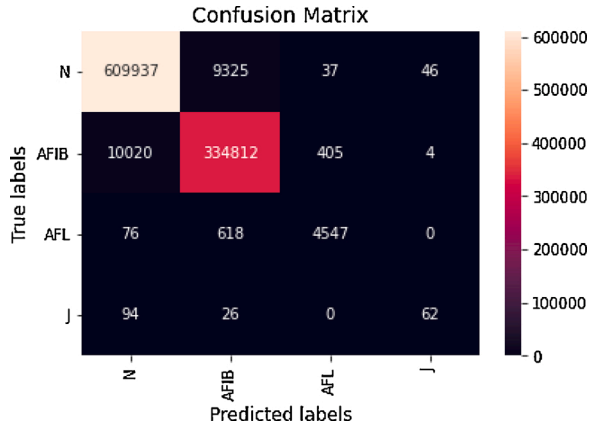


Fig. 6. The overall confusion matrix for the CNN-LSTM model with FL.

$$\text{specificity} = \frac{TN}{TN + FP} \quad (4.3)$$

$$F1 - \text{score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (4.4)$$

In this way, given the above equations, the evaluation metrics for each one of the four classes can be calculated, as given by Table 2.

The selection of the focusing parameter  $\gamma$  is a really critical issue for the calculation of the FL. Table 3 presents the overall sensitivity and specificity of the proposed model for different values of the  $\gamma$  parameter. FL with  $\gamma = 0$  corresponds to the CE loss. As it can be noted, the model reaches its maximum sensitivity and specificity values for  $\gamma = 2$  and except for  $\gamma = 3$  all the other  $\gamma$  values give equal or better results than the CE loss regarding the network's performance.

Finally, a comparison between the evaluation metrics of the proposed model and recent studies for the automated detection of AFib using the MIT-BIH Atrial Fibrillation Database is presented in Table 4. The proposed CNN-LSTM model with FL achieves higher specificity (99.29%) than every other previous study of the last five years for the

**Table 2**  
The evaluation metrics for the CNN-LSTM model with FL.

classes	precision	recall	specificity	F1-score	number of beats
N	98.36%	98.48%	97.09%	98.42%	619345
AFIB	97.11%	96.98%	98.4%	97.05%	345241
AFL	91.14%	86.76%	99.95%	88.9%	5241
J	55.36%	34.07%	100%	42.18%	182

**Table 3**

Overall sensitivity and specificity of the proposed model for different  $\gamma$  parameters.

$\gamma$ parameter	0	0.5	1	2	3
sensitivity	97.63%	97.77%	97.22%	97.87%	97.63%
specificity	99.21%	99.26%	99.07%	99.29%	99.21%

**Table 4**

Comparison between the proposed model and relevant studies.

study	method	sensitivity	specificity
Islam et al. [65]	Affine normalization of RR intervals	96.39%	96.38%
Kennedy et al. [66]	RF	92.8%	98.3%
	KNN	68%	95.1%
Andersen et al. [67]	SVM with Inter Beat Intervals	96.81%	96.20%
Cui et al. [68]	RR intervals dissimilarity	97.04%	97.96%
Kumar et al. [69]	WT with RF	95.8%	97.6%
Xu et al. [49]	MFSWT and CNN	74.96%	86.41%
Faust et al. [70]	LSTM	98.32%	98.67%
Andersen et al. [71]	CNN and RNN	98.96%	86.04%
Czabanski et al. [72]	LSVM	98.94%	98.39%
<b>This work</b>	<b>CNN-LSTM with FL</b>	<b>97.87%</b>	<b>99.29%</b>

same dataset, while the sensitivity value that corresponds to the overall accuracy of the network is higher than most of the relevant studies (97.87%), but lower than a few studies of the last two years.

Nonetheless, the performance of a deep learning model depends both on its capability of predicting correctly the class in which the data belong to (sensitivity) and its capability of rejecting wrong classes (specificity). This practically means that some models can accurately predict the correct class for the majority of the data (TP, FP), but fail to reject the wrong classes for some of the data (TN, FP). For this reason, the two metrics, sensitivity and specificity, should be simultaneously high enough in order that the model performs well for unseen data.

The hybrid CNN-LSTM model has been selected after a series of experiments regarding the number or the type of layers that have been implemented. More specifically, it has been found that by omitting some of the network's layers, the training time could be reduced as the total network calculations were significantly fewer. Nonetheless, it was noted that by dropping layers, the average sensitivity and specificity has been decreased, namely by using only the CNN part of the network sensitivity and specificity values were 97.48% and 99.16%, and by using only the LSTM part of the network the above values were 96.30% and 98.77%, respectively. This means that the hybrid network outcompetes both versions as it combines their capabilities to reach its full potential, despite the fact that the training time is slightly reduced in both cases

(34 s and 33 s respectively). Conversely, by using more CNN layers the overall sensitivity and specificity remained stable and the training time increased to about 1 min per epoch.

In general, the achievements of the present study are the following: the hybrid CNN-LSTM approach provides the best combination of performance (sensitivity, specificity) in comparison with all previous relevant studies, the proposed model performs well for highly imbalanced datasets, focal loss function delivers better results than the classic cross-entropy function for ECG classification and the proposed method could be used for real-time arrhythmia detection as the prediction phase lasts only a few seconds.

On the other hand, the limitations of the present study could be summarized as follows: the proposed network was tested only on four beat types, classes AFL and J represent only an extremely small percentage of the total dataset and the model's training by using 10-fold cross validation is very time consuming and makes a demand for use of high-tech computers.

Although, the present study was concentrated to the detection of cardiovascular arrhythmias in ECG signals using only data from the MIT-BIH Atrial Fibrillation Database, in the future we intent to test the performance of the proposed method on new data from other databases as well.

## 5. Conclusions

The capability of CNN to extract the most significant features from a body of data and reduce their dimensionality at the same time can be harmoniously combined with the capability of the LSTM networks to recognize and memorize the long-term dependencies between the extracted data to design robust models for the automated detection of arrhythmias in a time-series ECG. In this study, a hybrid CNN-LSTM model that implements FL to deal with data imbalance was proposed in order to detect arrhythmias in ECG signals deriving from AFib patients.

To the best of our knowledge, this is the first study that combines the methodology of the CNN and LSTM network with FL for the automated detection of arrhythmias in ECG signals. The proposed model receives ECG beats as input and predicts accurately (sensitivity 97.87%, specificity 99.29%) the class to which each beat belongs to.

## Funding

This work was supported in part from the ICT-2013-10-611223 eHealth WELCOME project funded by the EU and the INTERREG-IPA-CBC-2014-2020 Cross4All project funded by the EU.

## CRediT authorship contribution statement

**Georgios Petmezas:** Methodology, Software, Writing - original draft. **Kostas Haris:** Methodology, Software, Writing - original draft. **Leandros Stefanopoulos:** Software, Visualization. **Vassilis Kilintzis:** Data curation, Validation. **Andreas Tzavelis:** Conceptualization, Writing - original draft, Writing - review & editing. **John A Rogers:** Supervision, Writing - review & editing. **Aggelos K Katsaggelos:** Methodology, Writing - review & editing. **Nicos Maglaveras:** Supervision, Conceptualization, Writing - review & editing.

## Declaration of Competing Interest

The authors report no declarations of interest.

## References

[1] Institute of Medicine (US) Committee on Preventing the Global Epidemic of Cardiovascular Disease: Meeting the Challenges in Developing Countries, Promoting Cardiovascular Health in the Developing World: A Critical Challenge to Achieve Global Health, in: V. Fuster, B.B. Kelly (Eds.), *Epidemiology of*

*Cardiovascular Disease*, Vol. 2, National Academies Press (US), Washington (DC), 2010. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK45688/>.

[2] G. Aarabi, R. Schnabel, G. Heydecke, U. Seedorf, Potential Impact of Oral Inflammations on Cardiac Functions and Atrial Fibrillation, *Biomolecules* 8 (3) (2018) 66, <https://doi.org/10.3390/biom8030066>.

[3] B.R. Weil, C. Ozcan, Cardiomyocyte Remodeling in Atrial Fibrillation and Hibernating Myocardium: Shared Pathophysiologic Traits Identify Novel Treatment Strategies? *BioMed Research International* (2015) 1–8, <https://doi.org/10.1155/2015/587361>.

[4] R. Chebbout, E.G. Heywood, T.M. Drake, J.R.L. Wild, J. Lee, M. Wilson, M.J. Lee, A systematic review of the incidence of and risk factors for postoperative atrial fibrillation following general surgery, *Anaesthesia* 73 (4) (2017) 490–498, <https://doi.org/10.1111/anae.14118>.

[5] M. Anselmino, A. Battaglia, C. Gallo, S. Gili, M. Matta, D. Castagno, et al., Atrial fibrillation and female sex, *Journal of Cardiovascular Medicine* 16 (12) (2015) 795–801, <https://doi.org/10.2459/jcm.0000000000000239>.

[6] G.V. Naccarelli, H. Varker, J. Lin, K.L. Schulman, Increasing Prevalence of Atrial Fibrillation and Flutter in the United States, *The American Journal of Cardiology* 104 (11) (2009) 1534–1539, <https://doi.org/10.1016/j.amjcard.2009.07.022>.

[7] P. Sodmann, M. Vollmer, N. Nath, L. Kaderali, A convolutional neural network for ECG annotation as the basis for classification of cardiac rhythms, *Physiological Measurement* 39 (10) (2018) 104005, <https://doi.org/10.1088/1361-6579/aae304>.

[8] P. Hao, X. Gao, Z. Li, J. Zhang, F. Wu, C. Bai, Multi-branch Fusion Network for Myocardial Infarction Screening from 12-lead ECG Images, *Computer Methods and Programs in Biomedicine* (2019) 105286, <https://doi.org/10.1016/j.cmpb.2019.105286>.

[9] G.A. Tadesse, T. Zhu, Y. Liu, Y. Zhou, J. Chen, M. Tian, D. Clifton, Cardiovascular disease diagnosis using cross-domain transfer learning, 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2019), <https://doi.org/10.1109/embc.2019.8857737>.

[10] A. Khazaei, A. Ebrahimzadeh, Heart Arrhythmia Detection using support vector machines, *Intelligent Automation & Soft Computing* 19 (1) (2013) 1–9, <https://doi.org/10.1080/10798587.2013.771456>.

[11] J. Park, K. Lee, K. Kang, Arrhythmia detection from heartbeat using k-nearest neighbor classifier, 2013 IEEE International Conference on Bioinformatics and Biomedicine (2013) 15–22, <https://doi.org/10.1109/bibm.2013.6732594>.

[12] N. Maglaveras, T. Stamkopoulos, K. Diamantaras, C. Pappas, M. Strintzis, ECG pattern recognition and classification using non-linear transformations and neural networks: A review, *International Journal of Medical Informatics* 52 (1–3) (1998) 191–208, [https://doi.org/10.1016/s1386-5056\(98\)00138-5](https://doi.org/10.1016/s1386-5056(98)00138-5).

[13] N. Maglaveras, T. Stamkopoulos, C. Pappas, M.G. Strintzis, An adaptive backpropagation neural network for real-time ischemia episodes detection: development and performance analysis using the European ST-T database, *IEEE Transactions on Biomedical Engineering* 45 (7) (1998) 805–813, <https://doi.org/10.1109/10.686788>.

[14] M. Coskun, Ö. Yildirim, A. Ucar, Y. Demir, An overview of popular deep learning methods, *European Journal of Technic* 7 (2) (2017) 165–176, <https://doi.org/10.23884/ejt.2017.7.2.11>.

[15] U. Erdenebayar, H. Kim, J.-U. Park, D. Kang, K.-J. Lee, Automatic Prediction of Atrial Fibrillation Based on Convolutional Neural Network Using a Short-term Normal Electrocardiogram Signal, *Journal of Korean Medical Science* 34 (7) (2019), <https://doi.org/10.3346/jkms.2019.34.e64>.

[16] K.-S. Lee, S. Jung, Y. Gil, H.S. Son, Atrial fibrillation classification based on convolutional neural networks, *BMC Medical Informatics and Decision Making* 19 (1) (2019), <https://doi.org/10.1186/s12911-019-0946-1>.

[17] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation* 9 (8) (1997) 1735–1780, <https://doi.org/10.1162/neco.1997.9.8.1735>.

[18] S.L. Oh, E.Y.K. Ng, R.S. Tan, U.R. Acharya, Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats, *Computers in Biology and Medicine* 102 (2018) 278–287, <https://doi.org/10.1016/j.compbiomed.2018.06.002>.

[19] M. Kim, B. Cao, T. Mau, J. Wang, Speaker-Independent Silent Speech Recognition From Flesh-Point Articulatory Movements Using an LSTM Neural Network, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (12) (2017) 2323–2336, <https://doi.org/10.1109/taslp.2017.2758999>.

[20] M. Sundermeyer, H. Ney, R. Schluter, From Feedforward to Recurrent LSTM Neural Networks for Language Modeling, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23 (3) (2015) 517–529, <https://doi.org/10.1109/taslp.2015.2400218>.

[21] S. Chauhan, L. Vig, Anomaly detection in ECG time signals via deep long short-term memory networks, 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (2015), <https://doi.org/10.1109/dsaa.2015.7344872>.

[22] G.B. Moody, R.G. Mark, A new method for detecting atrial fibrillation using R-R intervals, *Computers in Cardiology* 10 (1983) 227–230.

[23] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P.C. Ivanov, R. Mark, et al., PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation [Online]*, 101 (23) (2000) e215–e220.

[24] A.L. Valderrama, S.B. Dunbar, G.A. Mensah, Atrial Fibrillation, *American Journal of Preventive Medicine* 29 (5) (2005) 75–80, <https://doi.org/10.1016/j.amepre.2005.07.021>.

[25] M. Zoni-Berisso, F. Lercari, T. Carazza, S. Domenicucci, Epidemiology of atrial fibrillation: European perspective, *Clinical Epidemiology* 6 (2014) 213–220, <https://doi.org/10.2147/clep.s47385>.

[26] Y. Miyasaka, M.E. Barnes, B.J. Gersh, S.S. Cha, K.R. Bailey, W.P. Abhayaratna, et al., Secular Trends in Incidence of Atrial Fibrillation in Olmsted County,



- Minnesota, 1980 to 2000, and Implications on the Projections for Future Prevalence, *Circulation* 114 (2) (2006) 119–125, <https://doi.org/10.1161/circulationaha.105.595140>.
- [27] B.P. Krijthe, A. Kunst, E.J. Benjamin, G.Y.H. Lip, O.H. Franco, A. Hofman, et al., Projections on the number of individuals with atrial fibrillation in the European Union, from 2000 to 2060, *European Heart Journal* 34 (35) (2013) 2746–2751, <https://doi.org/10.1093/eurheartj/ehd280>.
- [28] S.S. Chugh, R. Havmoeller, K. Narayanan, D. Singh, M. Rienstra, E.J. Benjamin, et al., Worldwide Epidemiology of Atrial Fibrillation, *Circulation* 129 (8) (2014) 837–847, <https://doi.org/10.1161/circulationaha.113.005119>.
- [29] S. Westerman, N. Wenger, Gender Differences in Atrial Fibrillation: A Review of Epidemiology, Management, and Outcomes, *Current Cardiology Reviews* 15 (2) (2019) 136–144, <https://doi.org/10.2174/1573403x15666181205110624>.
- [30] J.O. Cerasuolo, L.E. Cipriano, L.A. Sposato, The complexity of atrial fibrillation newly diagnosed after ischemic stroke and transient ischemic attack, *Current Opinion in Neurology* 30 (1) (2017) 28–37, <https://doi.org/10.1097/wco.0000000000000410>.
- [31] M.K. Chung, M. Refaat, W.-K. Shen, V. Kutyifa, Y.-M. Cha, L. Di Biase, et al., Atrial Fibrillation, *Journal of the American College of Cardiology* 75 (14) (2020) 1689–1713, <https://doi.org/10.1016/j.jacc.2020.02.025>.
- [32] European Heart Rhythm A, European Association for Cardio-Thoracic S, A. J. Camm, P. Kirchhof, G.Y. Lip, U. Schotten, I. Savelieva, S. Ernst, I.C. van Gelder, N. Al-Attar, G. Hindricks, B. Prendergast, H. Heidbuchel, O. Alfieri, A. Angelini, D. Atar, P. Colonna, R. De Caterina, J. De Sutter, A. Goette, et al., Guidelines for the management of atrial fibrillation: The Task Force for the Management of Atrial Fibrillation of the European Society of Cardiology (ESC), *EP Europace* 12 (10) (2010) 1360–1420, <https://doi.org/10.1093/europace/euq350>.
- [33] V.K. Srivastava, D. Prasad, DWT-Based Feature Extraction from ecg Signal, *American J. of Eng. Research (AJER)* 2 (3) (2013) 44–50.
- [34] V.X. Afonso, W.J. Tompkins, T.Q. Nguyen, S. Luo, ECG beat detection using filter banks, *IEEE Transactions on Biomedical Engineering* 46 (2) (1999) 192–202, <https://doi.org/10.1109/10.740882>.
- [35] S. Osowski, L.T. Hoai, T. Markiewicz, Support Vector Machine-Based Expert System for Reliable Heartbeat Recognition, *IEEE Transactions on Biomedical Engineering* 51 (4) (2004) 582–589, <https://doi.org/10.1109/tbme.2004.824138>.
- [36] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, L. Sornmo, Clustering ECG complexes using Hermite functions and self-organizing maps, *IEEE Transactions on Biomedical Engineering* 47 (7) (2000) 838–848, <https://doi.org/10.1109/10.846677>.
- [37] D. Castro, P. Felix, J. Presedo, A Method for Context-Based Adaptive QRS Clustering in Real Time, *IEEE Journal of Biomedical and Health Informatics* 19 (5) (2015) 1660–1671, <https://doi.org/10.1109/jbhi.2014.2361659>.
- [38] J.G. Carbonell, R.S. Michalski, T.M. Mitchell, An Overview of Machine Learning, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), *Machine Learning. Symbolic Computation*, Springer, Berlin, Heidelberg, 1983.
- [39] I. Mrazova, M. Kukacka, Can Deep Neural Networks Discover Meaningful Pattern Features? *Procedia Computer Science* 12 (2012) 194–199, <https://doi.org/10.1016/j.procs.2012.09.053>.
- [40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444, <https://doi.org/10.1038/nature14539>.
- [41] D. Li, J. Zhang, Q. Zhang, X. Wei, Classification of ECG signals based on 1D convolution neural network, 2017 IEEE 19th International Conference on E-Health Networking, Applications and Services (Healthcom) (2017), <https://doi.org/10.1109/healthcom.2017.8210784>.
- [42] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J. E. Mietus, G.B. Moody, C.K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, PhysioNet, Components of a New Research Resource for Complex Physiologic Signals, *Circulation* 101 (23) (2003) 215–220.
- [43] G.B. Moody, R.G. Mark, The impact of the MIT-BIH Arrhythmia Database, *IEEE Engineering in Medicine and Biology Magazine* 20 (3) (2001) 45–50.
- [44] K. Ochiai, S. Takahashi, Y. Fukazawa, Arrhythmia Detection from 2-lead ECG using Convolutional Denoising Autoencoders, 2018.
- [45] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, A. Gertych, R.S. Tan, A deep convolutional neural network model to classify heartbeats, *Computers in Biology and Medicine* 89 (2017) 389–396, <https://doi.org/10.1016/j.combiomed.2017.08.022>.
- [46] B. Pourbabaee, M.J. Roshkthari, K. Khorasani, Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48 (12) (2018) 2095–2104, <https://doi.org/10.1109/tsmc.2017.2705582>.
- [47] G.B. Moody, A.L. Goldberger, S. McClennen, S.P. Swiryn, Predicting the Onset of Paroxysmal Atrial Fibrillation: The Competing in Cardiology Challenge 2001, *Computers in Cardiology* 28 (2001) 113–116.
- [48] T.J. Jun, H.M. Nguyen, D. Kang, D. Kim, D. Kim, Y. Kim, ECG arrhythmia classification using a 2-D convolutional neural network, *ArXiv* (2018) abs/1804.06812.
- [49] X. Xu, S. Wei, C. Ma, K. Luo, L. Zhang, C. Liu, Atrial Fibrillation Beat Identification Using the Combination of Modified Frequency Slice Wavelet Transform and Convolutional Neural Networks, *Journal of Healthcare Engineering* (2018) 1–8, <https://doi.org/10.1155/2018/2102918>.
- [50] S. Hochreiter, The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06 (02) (1998) 107–116, <https://doi.org/10.1142/s0218488598000094>.
- [51] J. Gao, H. Zhang, P. Lu, Z. Wang, An Effective LSTM Recurrent Network to Detect Arrhythmia on Imbalanced ECG Dataset, *Journal of Healthcare Engineering* (2019) 1–10, <https://doi.org/10.1155/2019/6320651>.
- [52] Ö. Yildirim, A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification, *Computers in Biology and Medicine* 96 (2018) 189–202, <https://doi.org/10.1016/j.combiomed.2018.03.016>.
- [53] B.-H. Kim, J.-Y. Pyun, ECG Identification For Personal Authentication Using LSTM-Based Deep Recurrent Neural Networks, *Sensors* 20 (11) (2020) 3069, <https://doi.org/10.3390/s20113069>.
- [54] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research* 16 (2002) 321–357, <https://doi.org/10.1613/jair.953>.
- [55] Ö.R. Sanabila, I. Kusuma, W. Jatmiko, Generative oversampling method (GenOMe) for imbalanced data on apnea detection using ECG data, 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (2016) 572–577, <https://doi.org/10.1109/icacsis.2016.7872805>.
- [56] K.N.V.P.S. Rajesh, R. Dhuli, Classification of imbalanced ECG beats using re-sampling techniques and AdaBoost ensemble classifier, *Biomedical Signal Processing and Control* 41 (2018) 242–254, <https://doi.org/10.1016/j.bspc.2017.12.004>.
- [57] N. Betancourt, M. Flores-Calero, C. Almeida, ECG Denoising by using FIR and IIR Filtering Techniques, *Proceedings of the 2019 11th International Conference on Bioinformatics and Biomedical Technology* (2019) 111–117, <https://doi.org/10.1145/3340074.3340088>.
- [58] C. Saxena, A. Sharma, R. Srivastav, H.K. Gupta, Denoising of ECG Signals Using FIR & IIR Filter: A Performance Analysis, *International Journal of Engineering & Technology* 2 (2018) 51–58, <https://doi.org/10.14419/ijet.v7i4.12.20982>.
- [59] S.V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, Wiley, Hoboken, NJ, United States, 2009, <https://doi.org/10.1002/9780470740156>.
- [60] S. Butterworth, *On the Theory of Filter Amplifiers*, *Experimental Wireless and the Wireless Engineer* 7 (1930) 536–541.
- [61] D.L. Donoho, I.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika* 81 (3) (1994) 425–455, <https://doi.org/10.1093/biomet/81.3.425>.
- [62] D.L. Donoho, De-noising by soft-thresholding, *IEEE Transactions on Information Theory* 41 (3) (1995) 613–627, <https://doi.org/10.1109/18.382009>.
- [63] M. Alfaouri, K. Daqrouq, ECG Signal Denoising By Wavelet Transform Thresholding, *American Journal of Applied Sciences* 5 (3) (2008) 276–281, <https://doi.org/10.3844/ajassp.2008.276.281>.
- [64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal Loss for Dense Object Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2) (2020) 318–327, <https://doi.org/10.1109/tpami.2018.2858826>.
- [65] M.S. Islam, N. Ammour, N. Alajlan, H. Aboalsamh, Rhythm-based heartbeat duration normalization for atrial fibrillation detection, *Computers in Biology and Medicine* 72 (2016) 160–169, <https://doi.org/10.1016/j.combiomed.2016.03.015>.
- [66] A. Kennedy, D.D. Finlay, D. Guldenring, R.R. Bond, K. Moran, J. McLaughlin, Automated detection of atrial fibrillation using R-R intervals and multivariate-based classification, *Journal of Electrocardiology* 49 (6) (2016) 871–876, <https://doi.org/10.1016/j.jelectrocard.2016.07.033>.
- [67] R.S. Andersen, E.S. Poulsen, S. Puthusserypady, A novel approach for automatic detection of Atrial Fibrillation based on Inter Beat Intervals and Support Vector Machine, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2017) 2039–2042, <https://doi.org/10.1109/embc.2017.8037253>.
- [68] X. Cui, E. Chang, W.-H. Yang, B. Jiang, A. Yang, C.-K. Peng, Automated Detection of Paroxysmal Atrial Fibrillation Using an Information-Based Similarity Approach, *Entropy* 19 (12) (2017) 677, <https://doi.org/10.3390/e19120677>.
- [69] M. Kumar, R.B. Pachori, U. Rajendra Acharya, Automated diagnosis of atrial fibrillation ECG signals using entropy features extracted from flexible analytic wavelet transform, *Biocybernetics and Biomedical Engineering* 38 (3) (2018) 564–573, <https://doi.org/10.1016/j.bbe.2018.04.004>.
- [70] O. Faust, A. Shenfield, M. Kareem, T.R. San, H. Fujita, U.R. Acharya, Automated detection of atrial fibrillation using long short-term memory network with RR interval signals, *Computers in Biology and Medicine* 102 (2018) 327–335, <https://doi.org/10.1016/j.combiomed.2018.07.001>.
- [71] R.S. Andersen, A. Peimankar, S. Puthusserypady, A deep learning approach for real-time detection of atrial fibrillation, *Expert Systems with Applications* 115 (2019) 465–473, <https://doi.org/10.1016/j.eswa.2018.08.011>.
- [72] R. Czabanski, K. Horoba, J. Wroble, A. Matonia, R. Martinek, T. Kupka, et al., Detection of Atrial Fibrillation Episodes in Long-Term Heart Rhythm Signals Using a Support Vector Machine, *Sensors* 20 (3) (2020) 765, <https://doi.org/10.3390/s20030765>.