

Article

BioECG: Improving ECG Biometrics with Deep Learning and Enhanced Datasets

Paloma Tirado-Martin *  and Raul Sanchez-Reillo 

University Group for Identification Technologies (GUTI), Carlos III University of Madrid, 28912 Leganés, Spain; rsreillo@ing.uc3m.es

* Correspondence: ptirado@ing.uc3m.es

Abstract: Nowadays, Deep Learning tools have been widely applied in biometrics. Electrocardiogram (ECG) biometrics is not the exception. However, the algorithm performances rely heavily on a representative dataset for training. ECGs suffer constant temporal variations, and it is even more relevant to collect databases that can represent these conditions. Nonetheless, the restriction in database publications obstructs further research on this topic. This work was developed with the help of a database that represents potential scenarios in biometric recognition as data was acquired in different days, physical activities and positions. The classification was implemented with a Deep Learning network, BioECG, avoiding complex and time-consuming signal transformations. An exhaustive tuning was completed including variations in enrollment length, improving ECG verification for more complex and realistic biometric conditions. Finally, this work studied one-day and two-days enrollments and their effects. Two-days enrollments resulted in huge general improvements even when verification was accomplished with more unstable signals. EER was improved in 63% when including a change of position, up to almost 99% when visits were in a different day and up to 91% if the user experienced a heartbeat increase after exercise.

Keywords: ECG biometrics; Deep Learning; pattern recognition; human verification; Convolutional Neural Networks; Long-Short Term Memory

Citation: Tirado-Martin, P.; Sanchez-Reillo, R. BioECG: Improving ECG Biometrics with Deep Learning and Enhanced Datasets. *Appl. Sci.* **2021**, *11*, 5880. <https://doi.org/10.3390/app11135880>

Academic Editors: Hugo Pedro Proença, Carmen Sánchez Ávila, Mariano López-García and Enrique Cantó Navarro

Received: 25 May 2021
Accepted: 19 June 2021
Published: 24 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Deep Learning techniques and tools have experienced noticeable and rapid improvements in the last decade. The concept of approaching analytical problems with analogies to the human brain was first published in the late 1950s [1]. However, deep architectures started getting attention in 2006 [2] after Hinton et al. succeeded in implementing the first fast Deep Learning algorithm [3]. Since then, Deep Learning approaches have provided great performances in pattern recognition. Medicine, Economics, and Robotics are only a few of the fields where these techniques can be applied.

Biometrics is a field heavily related to patterns. Therefore, it has been highly affected by these sophisticated algorithms. Some of the most traditional modalities are based on data that barely varies through time such as iris, fingerprint, or facial biometrics. However, the capability of identifying patterns in time-inconsistent signals has widened the possibilities of developing and improving new modalities. Behavioral biometrics such as gait or keystroke have been researched thanks to Deep Learning algorithms.

The Electrocardiogram (ECG) signal provides information about the electrical function of the human heart, usually measured by an electrocardiograph. The different combinations of sensor placements provide slight variations which allow enhancing specific parts of the waveform. The most common usage in health is the 12 leads, which are classified into limb and chest leads. These measurements have a general behavior as they are periodic and have recurrent patterns [4]. The most important part of the ECG is called the QRS complex as it is formed by waveforms Q, R and S. Usually it is extended to P-QRS-T, as it is surrounded by P and T waves, like those in Figure 1.

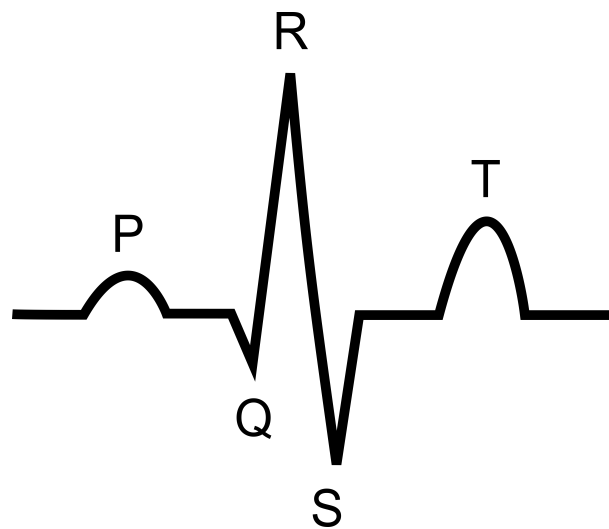


Figure 1. Normal Sinus Rhythm waveform and some of its relevant points.

Despite their generalities, these signals still present interindividual variability [5], making individual identification possible. In addition, it has been proven that the long-term intraindividual variability is comparable to day-to-day variability [6], allowing comparisons in different periods, even considering verifications with more than one-year separation [7]. Moreover, ECGs can be obtained from every live being, providing life-proof detection. These signals are difficult to fake, as they represent the functioning of an organ, which is involuntary and its activity cannot be accessed easily. These characteristics gift ECGs with great potential for biometrics and have been studied in this field since the early 2000s [8].

The present work deals with data acquired under different conditions related to time, position, and heart rate, as the process took place on two different days, with sitting and standing positions and after increasing the heartbeat frequency with exercise. The classification was achieved by using a combination of Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM). This choice helped in using the minimum data preprocessing and delivered information on how these techniques generalize when employing more realistic signals. The influence of factors such as the type of enrollment (one or two days) or length were observed and discussed through the hyperparameter optimization. A final configuration was set, providing general improvements, especially in those conditions related to unstable signals.

2. Related Work

Due to the general interest in medical research, there exists several public ECG databases, available in Physionet [9]. Some of the most important ones were developed before ECG was used in biometrics. However, they have been relevant at the early stages of this modality. The most popular ones are the MIT-BIH Normal Sinus Rhythm (MIT-BIH NSR) and PTB [10]. However, as the main focus was related to healthcare, the acquisition does not consider concerning issues in human recognition. Some issues that affect the field are related to variation between days, data quality, sensor placement, complex and expensive sensors, or being capable of performing under different environmental conditions. Specifically, the ECG can be affected by the user's physical activity before or during the acquisition process, as the heart rate fluctuates, stretching or widening the signal. The ECG-ID database [11] is the only public database collected with an on-the-person sensor that considered some of these issues. The ECG-ID database was formed by shorter signals with different physical positions for the subject: sitting and free movement. These features made a more realistic biometric environment, as it required shorter acquisition and included the possibility of physical movement. Since year 2005 more databases have been acquired for research in biometrics and on-the-person sensors, although they remain private [12]. Databases such as those in References [13,14] collected information from 502 and 269 users

respectively, starting with 10 s resting signals. The great number of users, as well as the type acquisition, situated these data closer to realistic biometric environments. However, long-term variations were not considered as those in the clinical database in Reference [15], which was formed by 10 s sessions of 460 patients, where the verification phase was carried out with at least one year difference.

The measurement of the heart activity presents several noise sources such as the baseline wander (0.2–0.5 Hz), power-line interferences (50–60 Hz), and muscle artifacts related to muscle movement (around 100 Hz). Discrete Cosine Transforms (DCT) [16] and Convolutional Neural Networks (CNN) [17] are two of the approaches to reduce the mentioned noise. However, the most frequent in literature is applying band-pass filters [18]. Feature extraction is usually applied after signal conditioning, where approaches are classified in two clear strategies: fiducial and nonfiducial features. Fiducial features are based on measuring different parts of the signal such as the amplitude of the R-peak of the width of the QRS complex, requiring a precise detection of these points, which is complicated to achieve. On the other hand, nonfiducial features are based on measurements or transformations that are only applied directly to the signal, without identifying specific points.

After preparing the data, training the classification model is the next step. Support Vector Machines (SVMs) and k-Nearest Neighbour (k-NN) algorithms are frequently applied in this stage for ECG biometrics [10] and other similar fields related to pattern recognition. However, the tendency is shifting to Neural Networks (NN), as they generalize properly from raw data, requiring less complex preprocessing techniques and achieving better results [19]. Thus, NN helped in simplifying the fiducial point detection to the point where, in some cases, it becomes expendable. Common machine learning techniques such as Multilayer Perceptron (MLP) have been applied to classification, but the higher complexity of Deep Neural Networks (DNN) is currently a common approach at this stage.

In Deep Learning, CNNs are commonly applied in research related to images. Some works use them as the main feature extraction tool in tasks, such as handwritten digit recognition [20]. However, they are also applied to problems related to 1D signals such as voice, where they contribute to a fast feature learning [21]. RNN are useful for time series, as they predict values based on the information provided in previous timestep. This characteristic feature makes RNN a suitable tool for recognition purposes. However, they present issues when long-term dependencies are required, due to the vanishing gradient problem. Thus, LSTM networks were designed, adding memory cells focused on keeping the long-term dependencies [22]. The combination of CNN and LSTM architectures is commonly used through literature in human recognition, where feature extraction and classification are carried out in the same network [23].

The constant research in mobile biometrics has also affected ECG biometrics in recent years. Approaches with hybrid features, combining fiducial and nonfiducial, complement each other, providing robustness [24]. Modifying the data acquisition to mobile-friendly steps, such as acquisition through the fingertips, also provides convenience and realistic characteristics required in research [25].

3. Materials and Methods

Following the general scheme of a biometric system, this section goes through the general preprocessing and classification stages, specifying details selected for the design. The applied database is then described with their correspondent consequences in the modeling. Finally, the used metrics are briefly detailed as a reference for the following section.

3.1. Preprocessing

The preprocessing stage transforms the obtained signals to increment their quality. In this work, the implemented method is the one applied in [26] and is developed in Matlab. The main goal is segmenting the QRS complex properly. First, the acquired signals got through a fifth-grade Butterworth filter, with low and high cut frequencies

of 1 and 35 Hz, to delete possible noise sources. Concerning the QRS segmentation, the main goal was detecting a reference point. Due to its prominence, this point was the R peak of the QRS complex. When first differentiated, R peaks are translated into more prominent local minima, which helps with an initial detection. To check the presence of outliers, adjacent complexes got compared by correlation. A second differentiation provided another shape confirmation. This detection process gave a reference point to segment the selected area, with length $rng1 + rng2$, where $rng1$ refers to the number of sample points to the left of the R peak and $rng2$ to the number of sample points to the right, including the reference point. The number of selected complexes, N , must be fixed by observation. No additional modification or feature extraction was applied to the data, as the goal is to avoid complex preprocessing.

3.2. Classification

This work aimed for user recognition with QRS complexes. Therefore, it was a supervised, multiclass classification problem. The focus was on using the combination of CNN and LSTM networks. However, these were not the only layers in the whole classification system, as some extra tools were required to improve its performance. The corresponding layers are represented in Figure 2 and are further discussed in this section.

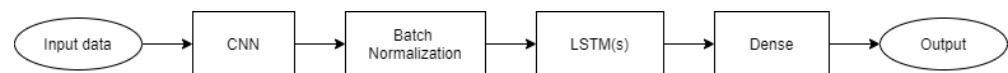


Figure 2. Scheme for the layers in the BioECG classification system.

3.2.1. Input Data

Every ECG signal resulted in a 2D-matrix with $[H_s \times W]$ dimensions, being $H_s = N_p$, where N_p was the number of detected peaks and $W = rng1 + rng2$, which represents the number of points selected left and right to the reference peak. At the same time, the number of ECG signals in a visit was determined by N_s and the number of different users by U . In the case that visits only collected one ECG signal, $N_s = U$. Generalizing, the total number of samples for all the database in a visit was $H_{db} = H_s \cdot N_s \cdot U$.

The selected data to fit the model were extracted from the enrollment data with the mentioned dimensions, called the development set. As validation was applied to this work, the development set was divided into two different sets: training and validation set. The samples in the development set were determined by a variable d , where $0 < d \leq 1$, which defined the proportion of available data to be part of the development set. H_d was the number of samples in the development set resulted from $H_d = H_{db} \cdot d$. Similarly, the training proportion was given by s_t , where $0 < s_t < 1$, and resulted in a training set size, $H_{train} = H_{db} \cdot s_t$ and analogously, the validation set size as $H_v = H_{db} \cdot (1 - s_t)$.

The training matrix was then fed into the network, named BioECG, with its corresponding label matrix. These labels were in the range $[0, U-1]$ and got one-hot encoded to help in the network calculations. This transformation resulted in a matrix with dimensions $[H_l \times W_l]$, where $H_l = N \times U \times N_s$ and $W_l = U$, as every column represents one class and there are as many classes as users in the database.

In addition, data was fed in batches for the whole process. This hyperparameter needed to be selected carefully, as dealing with big batches could lead to overfitting.

3.2.2. CNN

The purpose of this layer is summarizing the segmented data by extracting the most relevant features. It reduces the amount of data to interpret by the following LSTM units, easing the procedure and reducing its complexity [27].

The CNN has specific properties for one-dimensional signals, but the concept is similar to 2D CNN. The main difference is related to how the sliding window moves through the data. Two-dimensional sliding windows need to specify their width and height, as they slide horizontally and vertically. In the case of 1D convolutions, the only required

value is how many features are taken into consideration in every sliding iteration. The hyperparameters that affect the 1D CNN output size are:

- Kernel (k): the number of samples that are used in every iteration for the convolution.
- Filters (f): the number of sliding windows involved in the process, which translates into the number of extracted features.
- Strides (s): the number of positions the window slides each time.

The final dimensions are summarized in Figure 3, where b is the batch size, k the kernel size, f the number of extracted filters and how many units each filter strides. The output dimensions per batch correspond to a 3D matrix, as the process in every batch is done with f number of filters (b, o, f). The value o is the output width determined by Equation (1).

$$o = \frac{W - k}{s} + 1 \tag{1}$$

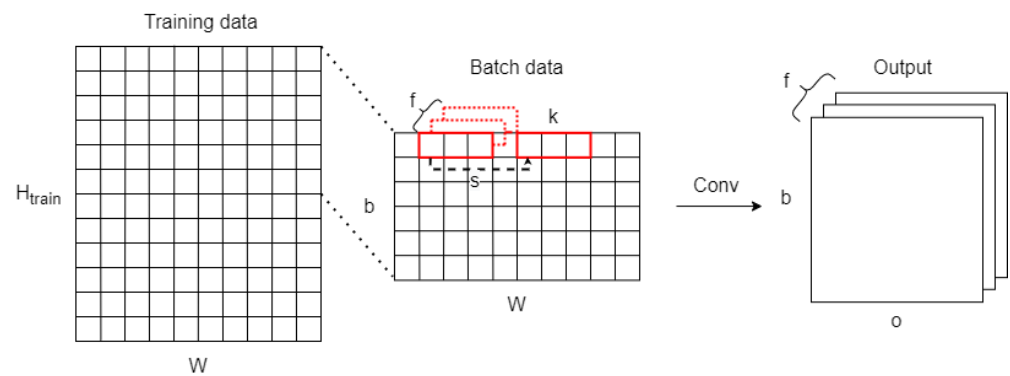


Figure 3. Scheme for CNN layer.

3.2.3. Batch Normalization

The Batch Normalization layer is added to help with convergence and learning between layers, when the input is fed in batches [28]. This layer keeps the same dimensions as those obtained in the CNN.

3.2.4. LSTM

LSTMs are a type of Recurrent Neural Network (RNN), as they also have chained recurrent modules. However, the different LSTM cells are more complex than those in standard RNNs. The specific structure is represented in Figure 4, where every rectangle represents a fully connected layer with their correspondent activations sigma (σ) and \tanh . Input data in the timestep t is represented by x_t . Similarly, the current cell state and outputs are represented by C_t and h_t .

The current cell state, C_t , depends on minor linear interactions related to the previous cell state, C_{t-1} . The LSTM gates are formed by a sigmoid (σ) layer and a pointwise multiplication so the outputs are kept between 0 (discarded information) and 1 (valid information). The forget gate, f_t , operation is in Equation (2), where W_f represents the correspondent weight matrix for that gate and stays unaltered through time. This calculation determines which information is required to be kept based on the previous output and the current information, plus b_f which represents a bias. Similarly, i_t is obtained with the same process and different weights and bias, W_i and b_i , as seen in Equation (3). This gate is known as the input gate; it selects which values get updated. The output i_t gets combined with a vector of candidate values, \tilde{C}_t , obtained with a \tanh layer, weights W_C and bias b_C , as observed in Equation (4).

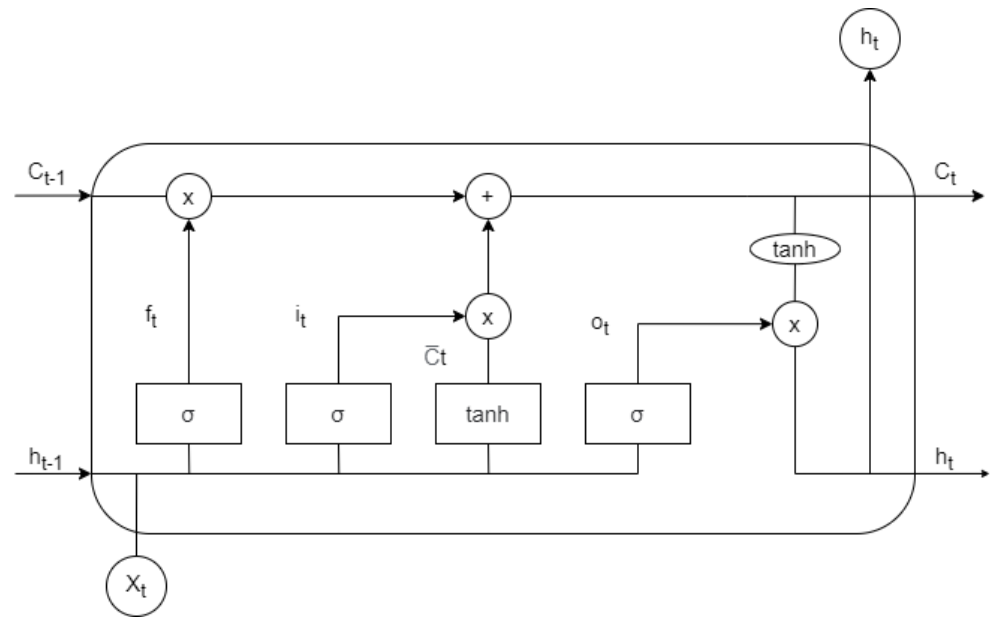


Figure 4. LSTM cell composition.

The previous cell state C_{t-1} updates resulting in Equation (5). C_t then a \tanh pushes it to values between -1 and 1 before getting multiplied by the output of another sigmoid gate. This part leads to the final output as formulated in Equations (6) and (7), using weights W_o and bias b_o . This process is done recurrently as many times as timesteps there are.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\bar{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t \times C_{t-1} + i_t \times \bar{C}_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \times \tanh(C_t) \quad (7)$$

To implement a multilayered LSTM, the output sequence of the LSTM cell in a given timestep, h_t , is returned and fed into the next layer. Figure 5 is an unrolled two-layered LSTM, where T represents the maximum number of timesteps. The last layer does not require retrieving all the hidden cell outputs but only the output in the last timestep. In the case of Figure 5, the final output corresponds to h'_T .

As a result, there are two hyperparameters to set:

- Hidden neurons (n): number of hidden neurons in the LSTM cell gates.
- Hidden layers (L): number of LSTM layers to connect.

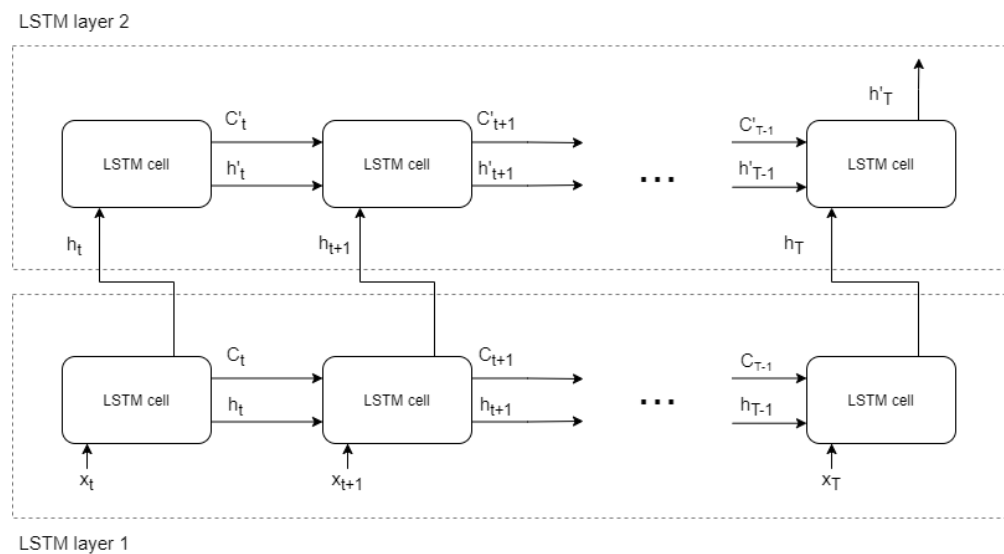


Figure 5. Scheme for an unrolled LSTM with two layers.

3.2.5. Dense

The previous dimensions need to be turned into results associated with every label in training. The output of the LSTM block is fed into a densely connected layer, with as many layers as labels the training data provided. This value has been previously referred to as U , as it corresponds to the number of users in the database. Adding a softmax activation in this layer allows obtaining results from 0 to 1, corresponding to probabilities. As a result, every batch has dimensions (b, U) as the final layer for every batch.

3.2.6. Output

When training the output matrix has dimensions (H_{train}, U) after proceeding with every batch. The values in every column represent the probability that the input sample i th, where $0 \leq i < H$, corresponds to label j th, where $0 \leq j < U$.

Analogously, in the case of testing, the output dimensions are (H_{test}, U) where H_{test} is the total number of test samples.

3.3. Database

The employed database included several considerations to fulfill the ISO 19795 requirements [29]. Data had to be representative of the target application. This collection process considered the nature of the ECG signal and how users deal with a biometric system. Based on these goals, data was acquired with the following criteria:

- Long time variation: data was collected in two different days (D1 and D2), with a minimum separation of 15 days. Two of the scenarios must be the same in both days to observe long-term variation.
- Short time variation: data was also collected twice every day in different visits (V1 and V2) to observe variations over short periods.
- Position variation: part of the users' data was acquired when the user was standing up, instead of sitting down. It allowed observing variability between these positions.
- Heart rate variation: part of the users' data was also acquired after increasing their heart rate to at least 130 beats per minute (bpm). For this purpose, the heart rate was monitored while doing exercise on a stepper. Once the heart rate reached 130 bpm or more, the user stopped and the data was collected while standing.

The collection was achieved with a professional medical sensor to provide higher signal quality. It allowed for the reduction of noise, even if it was not optimal for biometric applications. This decision was taken to reach results that were as unbiased to signal quality as possible, so they could be associated to the different scenarios. The data was

collected with a Biopac MP150 system with a 1000 Hz sample frequency, as described in Reference [26]. Only the type I lead was acquired, as it only involves the arms for measuring, which is appropriate to biometric applications because it only involves right and left arms.

Initially, measurements only considered short-term and long-term variations for 50 users. However, it was extended in a similar way to include position and heart rate variations. In total, both configurations involved 105 healthy users. Both configurations had the same methodology in relation to when the data was acquired: two visits per day on two different days. What changes between them is the number of users and the scenarios the users are in the second visit of every day. These variations are summarized in Table 1. Each visit records 5 signals of 70 s duration per signal with a 15 s posture adjustment between every recording.

Table 1. Selected database visit distribution, where the number after D represents the day, and the one after V, the visit in that day. The different scenarios are represented as R for resting, sitting down; S for resting, standing up and Ex for after exercise (average of 130 bpm). Each subset is named S1 or S2 for easier identification.

Set Name	Number of Users	D1V1	D1V2	D2V1	D2V2
S1	50	R	R	R	R
S2	55	R	S	R	Ex

The employed database remains private, due to the General Data Protection Regulation (GDPR) by the European Government. The law started its implementation in 2018 and considers that biometric data is sensitive. The GDPR takes into consideration the potential need to use this data in research but demanding some specific conditions. As this database was collected before the GDPR implementation, it does not fulfill the legal criteria to get published.

3.4. Modeling

This section discusses further details that need to be specified when fitting a model, as several features took a great part in the final process after determining the network structure. Specific dimensions related to the applied database are mentioned.

3.4.1. Input Data

A normal QRS has a duration up to 0.12 s [30]. Given the sampling frequency in the database, *rng1* and *rng2* were set to 100. This implies that the whole segmentation has a total temporal length of 0.2 s, where the R peak is in the 101th position. The duration was determined in order to ensure the whole consideration of the QRS complex. *U* then depends on the data scenario configuration in the database, where U_1 is the number of users in S1 and U_2 in S2. Each user provided $N_p = 50$ per signal, and 5 signals per visit so $N_s = 5$, concluding in 250 samples per user and visit. It finally translated into an available data dimension of (250, 200) per user and visit. The final development size still depended on the parameter *d*. However, as a common practice in Deep Learning, $s_t = 0.8$ as the training set is the 80% of the available data and validation the remaining 20%.

3.4.2. Classifier

Aiming for simplicity, the CNN was set to one layer with ReLU activation. The LSTM part of the classification stage only dealt with parameters *n* and *L*. The network was provided with values from 1 to 3, where the *n* was doubled in the first layer independently of *L*. The output dimensions between layers and the design relied on the circumstances, considering the different *l*th layers, where $1 \leq l \leq L$:

- $L > 1$ and $l = 1$, the output dimensions were $(b, o, 2n)$, as the number of hidden neurons is doubled.

- $L > 1$ and $1 < l < L$, retrieves dimensions of (b, o, n) .
- $L > 1$ and $l = L$, resulted in (b, n) dimensions, as it was the last layer.
- $L = 1$, the only layer was also the output layer. The output had dimensions $(b, 2n)$.

3.4.3. Output

In training, the final output consisted on a matrix with dimensions (H_{train}, U_1) when using S1 or (H_{train}, U_2) with S2. For testing, the dimensions were (H_{test}, U_1) and (H_{test}, U_2) respectively. The values in every column represented the probability that the input sample i th, where $0 \leq i_{th} < H_{test}$, corresponded to label j th, where $0 \leq j_{th} < U$.

3.4.4. Epochs and Early Stopping

The network iterates through the training data as many epochs as required. The number of epochs has a great impact on the model fitting. Having a value that is too high could lead to overfitting or underfitting if the opposite. To avoid choosing a general value, early stopping was used as a callback. It stops the fitting process once certain criteria are met. In this case, the process is stopped once the loss did not improve through a specific number of epochs, called patience. This system had a maximum number of epochs of 500 and patience of 20.

3.4.5. Loss Function

Then, the results are typically evaluated through losses. As the problem was multiclass, the target function was the cross-entropy loss, which is commonly used in literature. Faster training and improved generalization are advantages in comparison to the performance with classic losses, such as the sum-of-squares [31].

3.4.6. Optimizer and Learning Rate

A common approach for the optimizer selection is using Adam [32]. It has been demonstrated to outperform classic algorithms such as Root Mean Squared Propagation (RMSprop), even though they are based on the same stochastic gradient descent (SGD) optimization [33]. This algorithm depends on a learning rate (η), which helps to determine how high or low the updated weights are going to be. In this work, this value was set to 0.001.

3.4.7. Hyperparameter Optimization

The hyperparameter optimization was implemented using Keras in Python 3. Initially the code was run in a i7-6700 CPU without dedicated GPU. After several experiments, an upgrade in terms of GPU was required and changed into a i9-9900K CPU with 16 GB RAM and Nvidia GeForce RTX 2080 Ti GPU. Considering the limitations in hardware, the process in the optimization has the following steps:

1. Fix hyperparameters only related to CNN (s , k and f) heuristically by observing validation results in training when $d = 0.5$, $b = 35$, $L = 2$ and $n = 32$.
2. Random Search with LSTM hyperparameters to reduce their dimensionality [34]. The iterations are fixed to 50.
3. Grid Search for the present hyperparameter values in the first three best options in Random Search.

Table 2 summarizes the starting hyperparameter values in the optimization. Values that only correspond to the CNN are specified in bold, as they were determined through Step 1 and remained constant for Steps 2 and 3. Remaining sets of possible values were not determined at this point, as they varied depending on the experiment in the following sections.

Table 2. Initial hyperparameter values for modeling. Single final values for k , s , and f were selected in Step 1 and specified in bold font. Values of b , L , and n were determined with Steps 2 and 3 depending on the d parameter.

Hyperparameter	Values
Batch size (b)	[20, 35, 50, 100]
Kernel size (k)	[3, 5, 7, 9]
Strides (s)	[3, 5, 7, 9]
Filters (f)	[8, 16 , 32, 64]
Activation	ReLU
CNN layers	1
LSTM layers (L)	[1, 2, 3]
LSTM neurons (n)	[16, 32, 64, 128]
Epochs	500
Patience	20
Optimizer	Adam
Learning rate (η)	0.001

3.5. Metrics

The performance was obtained after training the model. The new data got tested by using cross-validation with 5 folds to check its correct behavior. The last trained model was used for the evaluation. The performance is evaluated differently in biometrics, as it depends on the task the system is executing. It is necessary to distinguish between identification and verification [35]. However, even though identification metrics are also provided, the focus of this work is on verification.

3.5.1. Identification

Identification is a one-vs.-all comparison, where the provided test sample is compared against all the enrolled information, looking for the best match. This work calculates accuracy in this case, also named Identification Ratio (IDR) in literature. The comparison determines how likely it is for the new sample or samples to belong to each of the different labels that took part in model training. The correct result is the one with the highest probability, assigning a 1 and setting the rest to 0. As the database was already labeled, it allowed checking the correctness of the decision, obtaining its accuracy. The predicted label was the one with the highest probability and it was compared with the true label. Averaging the number of correct predictions gave this metric.

3.5.2. Verification

Verification is a one-vs.-one comparison, where the system must check the subject's identity. Again, the provided result is probabilistic, but the only result is one probability instead of one value per class. This forces the establishment of a rule to determine when to consider if the probability is high enough to verify or not. Setting a threshold solves the problem but also implies false negative and false positive errors. Their corresponding measurements are called False Non-Match Rate (FNMR) and False Match Rate (FMR), respectively. Their cutting point represents the minimum possible error, named Equal Error Rate (EER). The EER is theoretically calculated when FNMR = FMR. However, values from these curves are not continuous in real case scenarios, so the EER has to be estimated from the available data, as represented in Figure 6. The threshold values th_1 and th_2 give the pairs of values FNMR and FMR that are the closest. These two pairs of points allow for the characterization of both FNMR and FMR as lines to calculate the crossing point that estimates the EER value.

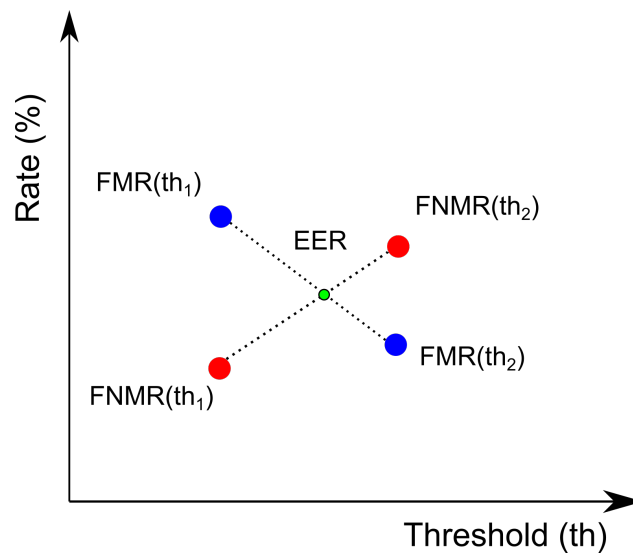


Figure 6. Graphic representation of the EER calculation from FNMR and FMR data.

The system’s performance in function of the decision threshold provides graphs called Detection Error Trade-off (DET). DET graphs provide more information about how restrictive or permissive the system can be based on its purpose.

4. Experimental Analysis

This work also studied how different training data and sizes affect the resulting performance. Based on this, the development set was obtained in two ways:

1. One-day data: the development set was only formed by data coming from D1V1. The parameter d provided information about the proportion of data in the development set, as explained in previous sections. The relationship between these values and the final samples per user and visit are summarized in Table 3.
2. Two-days data: the development set was formed equally by data from D1V1 and D2V1. The available data to form the development test was doubled with respect to one-day data, as it proceeded from two different visits. To make the results unbiased in relation to the amount of data in the development set, the total number of samples for a given value of d must be the same between one-day and two-days data. Therefore, the given total proportion of data to select in each visit is given by $d/2$ for each (D1V1 and D2V1). Note that in the case of two-days data, the final number of samples sometimes was not an integer so the value was rounded down. This fact affects the development, training, and validation set dimensions in one sample, which were not considered to affect the process. To avoid redundant information, these values are generalized in Table 3.

Table 3. Samples per user for development, training, and validation sets, based on d . In the case of one-day enrollment, it also represents the number of samples extracted from one visit, D1V1. In the case of two-days enrollment, the results could vary in one sample and are formed by data in D1V1 and D2V1 in the same proportion.

Dev. Set Proportion (d)	0.3	0.5	0.7	0.9
Dev. set samples (H_d)	75	125	175	225
Training samples (H_{train})	60	100	140	180
Validation set samples (H_v)	15	25	35	45

To achieve this, the performance evaluation was first achieved using D1V1 as the enrollment data, which forms the development set. Then, the same procedure was carried

out using two-days data. In addition, different values were given to the parameter d so that the development set never requires all the available data.

The parameter d provided the same number of samples in the development set, independently of using one-day or two-day enrollment. Varying d results in a certain number of samples per user as seen in Table 3, where in the case of using two-days data, half of the samples proceed from D1V1 and the remaining half, from D2V1. The tuning process was done independently for every development set size so that each different value of d provides its own set of hyperparameters.

The applied database recapitulated different types of variations, as referred to previously. These characteristics led to a specific result classification, as S1 only provided information about long and short time variations, whereas S2 also added variations in positions and heart rate. To reflect the effects of these variations, the verification data was obtained from different visits, as follows:

1. Same scenario.
 - (a) Variations in the same day: D1V1, D1V2 for S1. D1V1 for S2.
 - (b) Variations between days: D2V1, D2V2 for S1. D2V1 for S2.
2. Different scenario.
 - (a) Different position: D1V2 for S2.
 - (b) Different heart rate: D2V2 for S2.

Every one of the experiments was carried out twice, depending on the type of enrollment data: one-day or two-days.

4.1. Same Scenario

Both S1 and S2 provided information about variations within the same scenario, R. However, S1 provided 4 visits of R, whereas S2 only has 2. As a consequence, it is necessary to specify which set was used for the visits, as they may present differences.

4.1.1. Variations in the Same Day

The different results in the same day are summarized in Table 4; Table 4 shows one-day and two-days data in enrollment, respectively. Results were slightly better in the one-day enrollment, as all the data in the verification process belonged to the same day and visit, D1V1. The best results required a lower value of d than in the best results for the two-days enrollment. Even when requiring the same number of samples, the two-days enrollment had half of data related to D1V1, which results in requiring higher enrollment lengths to achieve equally good verification rates when verifying with D1V1. However, the number of available samples were split in half between D1 and D2 in a two-days enrollment, resulting in worse performances for visits in D1.

In the case of S2 for Table 4, $d = 0.9$ provided a very different result, as it was the only non-zero value. This could be a result of the individual tuning for every value of d , which could be not as accurate as in the rest of the cases. On the contrary, performances for D1V1 were better in S2 than in S1 in Table 4, both in identification and verification. In this case, the best metrics were achieved with $d = 0.9$, although results slightly differed with the remaining values.

Table 4. Identification and verification results for same day in S1 and S2 for scenario R. The best considered options for one-day and two-days enrollment are in bold font.

Dataset		S1				S2	
Visit		D1V1		D1V2		D1V1	
d	Metric	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
	One-day enrollment.	0.3	92.81	0	93.25	0	96.30
0.5		98.19	0	97.96	0	95.86	0
0.7		96.9	0	96.42	0	95.8	0
0.9		96.37	0	97.12	0	74.58	1.58
Two-days enrollment.	0.3	94.42	0.04	94.09	0.04	97.24	0
	0.5	90.17	0.45	89.58	0.57	91.64	0.03
	0.7	93.09	0.08	93.12	0.08	97.62	0
	0.9	94.55	0.04	95.2	0.04	96.39	0

4.1.2. Variation between Days

Results for R in different days were collected in Table 5. One-day data results are represented in Table 5. The decrease of accuracies and increment in EER show how different ECGs can be between days, even under the same scenario. Results in Table 5 belong to those with two-days development data. In this case, the improvement from one-day to two-days enrollment was relevant. From accuracies around 66–76% in Table 5 to values in Table 5 for $d = 0.3$. Maximum achieved accuracy reached 98.91% in $d = 0.7$, comparable to those with same-day scenarios, as seen in Table 4. In terms of verification, the EER decreased from 2–6.54% in one-day enrollment to values that ranged between 0% and 0.24% in two-days enrollment.

Table 5. Identification and verification results for different days in S1 and S2 for scenario R. The best considered options for one-day and two-days enrollment are in bold font.

Dataset		S1				S2	
Visit		D2V1		D2V2		D2V1	
d	Metric	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
	One-day enrollment	0.3	66.43	4	67.54	4	67.99
0.5		76.86	2	76.50	2.53	66.14	4.59
0.7		71.58	3.71	71.62	3.06	67.78	5.45
0.9		70.39	4	71.53	4	48.72	6.54
Two-days enrollment	0.3	92.54	0.04	92.27	0.04	98.81	0
	0.5	87.81	0.24	88.31	0.24	93.75	0.17
	0.7	94.83	0.12	94.95	0.12	98.91	0
	0.9	97.68	0	97.17	0	98.12	0

The different results between Tables 4 and 5 can be a product of the intraindividual variability between days, where doubling data related to D1V1 did not give enough information about the ECG variability in the long term. Moreover, comparing results in Table 5, results were noticeably better in all the values of d and in all the visits in D2 when applying a two-days enrollment. When $d = 0.9$, accuracies went from 48.72–71.53% to 97.17–98.12%.

4.2. Different Scenario

As seen in Table 1, the only set that provided data from different scenarios is S2. As a consequence, S2 was the only set of data required in the following experiments.

4.2.1. Different Position

The data that also considered the change of position was acquired in the same day as the enrollment but in a different visit. This implies that observed differences could only be related to the position and short time variation. Table 6 summarizes those results, where Table 6 relates to one-day enrollment. Even though test data was acquired on the same day as in Table 4, the change of position made the results decrease noticeably: almost a third in accuracy and going from almost ideal values up to 5.45% in EER in the worst-case scenario. This information confirms that changing the position affects the verification results even considering experiments in the same day.

Table 6. Identification and verification results for scenario S. The best considered options for one-day and two-days enrollment are in bold font.

		Dataset	S2	
		Visit	D1V2	
		Metric	Accuracy (%)	EER (%)
		d		
One-day enrollment		0.3	37.59	3.64
		0.5	67.32	5.45
		0.7	68.8	3.64
		0.9	63.78	4.54
Two-days enrollment		0.3	80.42	1.82
		0.5	73.88	3.64
		0.7	79.45	1.99
		0.9	85.14	1.28

Table 6 collected performances when changing into two-days enrollment. The results improved noticeably in comparison to those in Table 6. Even though the second day of data used in training did not provide information about the change of position, it has also helped to generalize in this case. It resulted in almost doubling the accuracy in the worst previous result, while decreasing the different EERs up to 50%.

4.2.2. Different Heart Rate

Performances for the change of heart rate are collected in Table 7. Both identification and verification presented a huge decrease in performance for one-day enrollment in Table 7. In contraposition to those results in Table 6, this verification data belongs to a different acquisition day, adding extra variations that may not be related to the heart rate. However, comparing results in the same scenario in Table 5 with those in Table 7 help in the assumption that the performance decrease is due to the heart rate variation.

Table 7. Identification and verification results for scenario Ex. The best considered options for one-day and two-days enrollment are in bold font.

		Dataset	S2	
		Visit	D2V2	
		Metric	Accuracy (%)	EER (%)
		d		
One-day enrollment		0.3	42.91	11.43
		0.5	44.36	9.83
		0.7	50.62	7.27
		0.9	29.26	14.16
Two-days enrollment		0.3	59.91	3.64
		0.5	58.58	3.64
		0.7	56.28	5.45
		0.9	58.04	3.90

On the other hand, Table 7 presents results with two-days enrollment. Results were still not good in terms of identification, remaining below 60%. However, in verification, EERs decreased noticeably. It was proved that adding an extra day of information for training improves the system noticeably in terms of verification, once it was compared to Table 7. In $d = 0.9$ EER dropped almost 10% and close to 7% in the case of $d = 0.3$.

4.3. Final Configuration

Considering the different results in the previous section, the solution to the final system's configuration was not unique. Depending on its purpose, some factors needed to be taken into account.

The enrollment process needed to be long enough to provide good information for the development set. However, if the enrollments were too long (i.e., a greater value of d), the user might get tired. Adding an extra day of acquisition has been proven to provide better results. Unfortunately, users are usually reticent to extend the enrollment process to several sessions. However, if the system was required to have higher performance, it may be worth the effort.

If the purpose of the system focused on fast recognition more than high performances, the enrollment process could get shorter and easier. It also would depend on the probability that users come up with different position or heart rate throughout recognition, e.g., members going in the gym may have different heartbeat frequency as when they go out, as they may have not been fully recovered yet.

Once these issues have been addressed, this work suggests one specific configuration as a trade-off choice, which was considered to provide good general verification results in all different scenarios. Doing a two-days enrollment is key for increasing the verification performance and even more if there are heart rate variations in the recognition process. The chosen enrollment size is $d = 0.5$ as it was a frequent value when obtaining the best discussed results. However, when it was not the best of all the proportions, it still performed properly while allowing to have a shorter enrollment process. Looking at Table 3, the enrollment required 125 samples per user between two-days. That means around 63 QRS samples per visit, which requires two ECG signal acquisition as 50 complexes are extracted from each one. It summarized in an enrollment process of a maximum duration of 140 s, as every 50 peaks requires 70 s of acquisition. Setting N_p to a higher number would allow for the enrollment of people with one signal acquisition, depending on the user's resting heart rate.

The hyperparameter tuning provided as the best configuration $b = 20$, $L = 2$, and $n = 64$. The tuning reached 151 epochs following the early stopping criteria. The mean time taken for training each fold in cross-validation was 147.7 s for the whole S2 dataset.

The system was trained according to these hyperparameters for S1 and S2 independently. The same process was achieved using S1 and S2 as a whole dataset too, providing heterogeneous samples as D1V2 and D2V2 are different scenarios. Verification performance results under these conditions are collected in Table 8. Using the S1 and S2 jointly in training allowed one to obtain the DETs in Figure 7, where Figure 7a,b represent results for one-day and two-days training, respectively. Analogously, Figures 8 and 9 are the DETs for S1 and S2 respectively, where Figures 8a and 9a are DETs for one-day data training and Figures 8a and 9b for two-days in training.

Table 8. EER (%) results for the final selected hyperparameters, with the different datasets (S1, S2, and S1 + S2) with one-day and two-days data in training. The EER variation between one-day and two-days is detailed as a percentage (%) in parenthesis when both data are non-zero.

Set	Visit Train	D1V1		D1V2		D2V1		D2V2	
		One-Day	Two-Days	One-Day	Two-Days	One-Day	Two-Days	One-Day	Two-Days
	S1 + S2	0	0	1.90	0.70 (−63)	4.72	0.01 (−99)	5.31	1.35 (−74)
	S1	0	0.45	0	0.57	2.30	0.24 (−89)	2.70	0.24 (−91)
	S2	0	0.03	5.27	3.93 (−25)	5.43	0.17 (−95)	10.13	3.64 (−64)

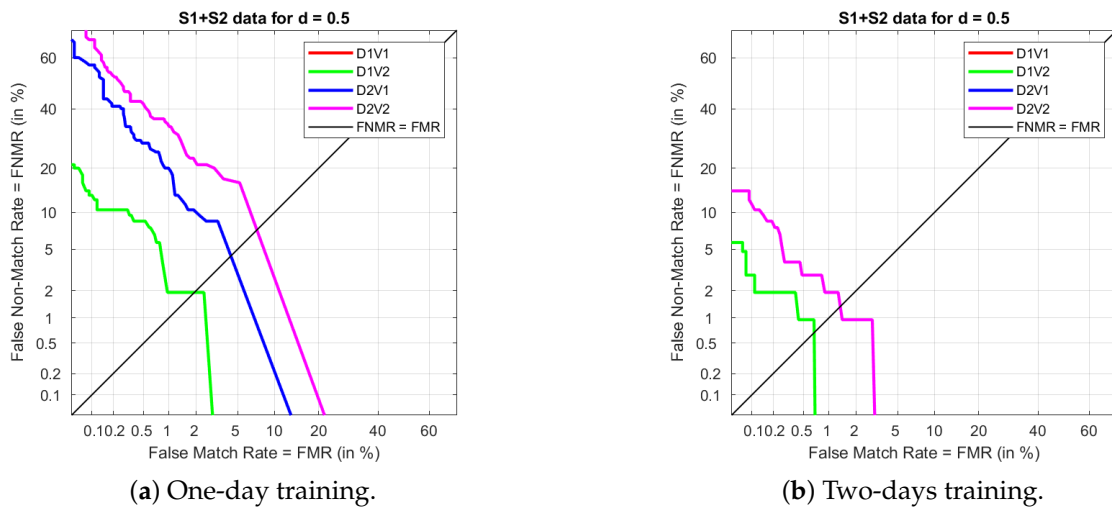


Figure 7. Performance results using data from S1 and S2.

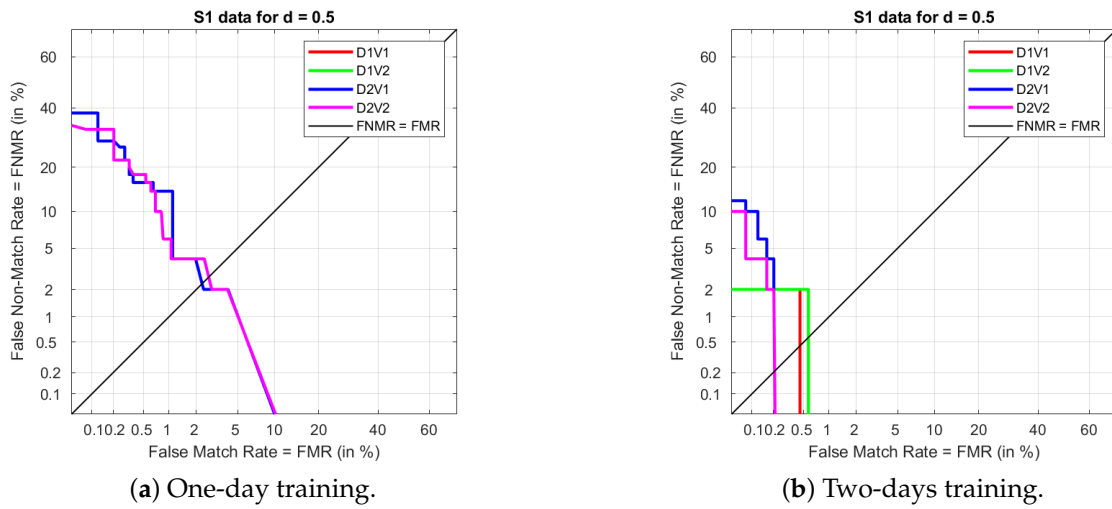


Figure 8. Performance results using data from S1.

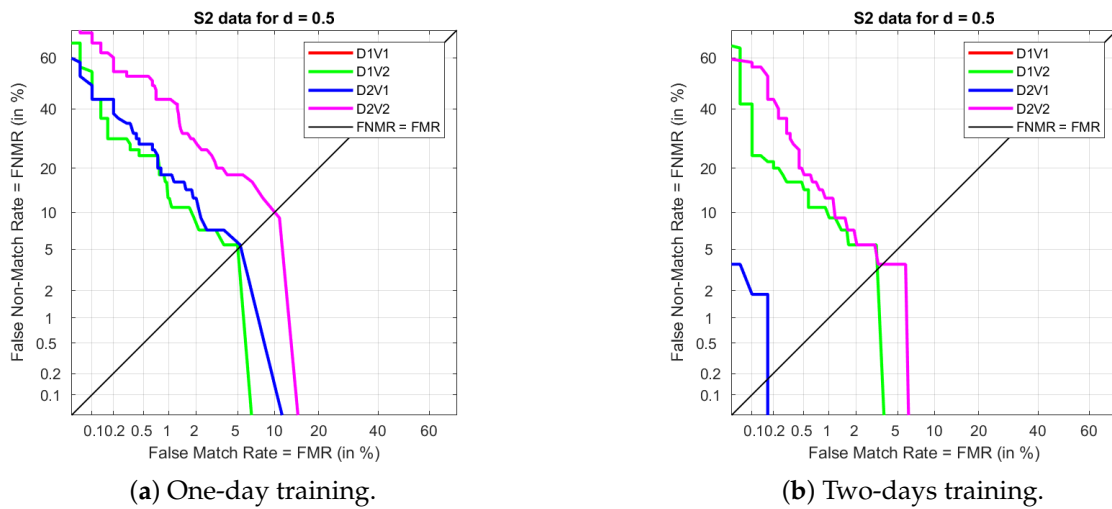


Figure 9. Performance results using data from S2.

Scenarios in S1 were constant, thus its results were the best ones in terms of EER. Using two-days in training decreased the metrics slightly on the first day of acquisition

but affected more positively when considering visits on a different day (i.e., from 2.70% to 0.24% for D2V2). When dealing with S2, results are always improved to those obtained under one-day enrollment. Results for D1V1 barely changed as they went from 0% to 0.03%. However, in the most dramatic change of scenario (D2V2 in S2), the EER decreased 64%. In S2 the scenario was the same but varying the day, and the EER got decreased by 91%. This result was a dramatic change in performance considering how the ECGs vary between days and physical activity. Using S1 and S2 jointly for training provided results that ranged between those in S1 and S2 independently, proving that two-days data enrollments could result in a great generalization even when dealing with heterogeneous data in recognition.

Comparison with Previous Works for the Same Database

Works related to the applied database were limited due to its privacy. However, a similar analysis was provided in a previous work [36], where subset S2 with MLP classifiers was employed. The enrollment was longer, with 187 QRS complexes for the best results, and it required the first differentiation of the QRS. The performance for S1 and S2 was also obtained in [26], and it required the Stationary Wavelet Transform (SWT) for the heartbeat classification and an extra Infinite Feature Selection (IFS). The differences and results are summarized in Table 9, including the best ones for the final configuration chosen in this work. Results for S2 improved noticeably when using the proposed two-days enrollment and classification compared to MLP, as the minimum EER goes from 2.69% in MLP to 0.03% in BioECG. Analogously, the worst-case goes from 4.71% to 3.93%. In the case of using a joint dataset with S1 and S2, results were improved using both types of enrollment. However, when using two-days enrollment, the lowest EER improved from 1.74% to 0% and the highest EER from 5.47% to 1.35%.

Table 9. Comparison with similar works that use the proposed database. Results are represented in a range of EER, where the first value is the result for verification with data in the same conditions as the enrollment. The range upper bound is the result for verification using data after exercise.

Reference	Dataset	Enrollment	Features	Classifier	EER (%)
Proposed work	S2	One-day Two-days	QRS complex	BioECG	0–10.13 0.03–3.93
[36]	S2	One-day	First differentiation QRS complex	MLP	2.69–4.71
Proposed work	S1 + S2	One-day Two-days	QRS complex	BioECG	0–5.31 0–1.35
[26]	S1 + S2	One-day	Heartbeat SWT + IFS	LDA	1.74–5.47

5. Discussion

Current approaches in ECG for human recognition are based on Deep Learning. These tools require representative data in order to perform at their maximum capacity. However, the selection of published ECG databases is limited, and most of them are not representative for a biometric scenario. The present work dealt with ECG data acquired in different positions, heartbeat frequency, and days in an attempt to further study its potential as a biometric signal. The results were achieved by applying a neural network as a classifier, BioECG, as it is a successful approach through literature. Through the tuning process, the influence of heart rate, position change, and long-term variations are also discussed.

The achieved tuning procedure was extensively done using different numbers of samples to observe its influence. However, results related to the number of samples in the enrollment were heterogeneous depending on the evaluated experiment. Choosing the final model to be taken requires knowledge about the system's purpose. A final configuration was specified as a compromise solution considering results in different scenarios and enrollment size. Based on that final tuning, this work has focused on one-day and two-days

enrollment processes while using the same number of total samples. The different scenarios proposed in the database gave information about how different day data help generalize unstable signals, too.

One-day enrollments provided good results when dealing with data acquired in the same conditions, such as day and physical scenario. EERs increased noticeably when calculated with data in different days, even considering the same resting state. Adding an extra day of enrollment affected verification in the same day, as a consequence of sharing the available data with an enrollment from another day. However, the error increase was low, even more if it is in exchange of the improvement under more complex conditions. As a consequence, results for visits in the second day were highly decreased up to 99% when dealing with resting scenarios. For high heart rate conditions, a second day of enrollment improved the EER from 64% to 91% with respect to one-day enrollment. This observation demonstrated how obtaining data in different days helped generalize more unstable signals.

Results for one day of enrollment were comparable to a previous work that dealt with the same database, with the difference that the present work required less data transformation in preprocessing. In the case of managing data in resting state, the results were improved even for enrollments in one day. After observing the results in BioECG with low signal processing, adding extra steps in this stage could provide more insight into this modality. More complexity in the R-peak detection including Deep Learning could help select better data. As the database was limited to two acquisition days, adding extra visits for different days should be considered. This addition could help in the assessment of the performance in data that does not belong to any of the days that take part in enrollment, avoiding any possible bias. Further research with portable devices should be taken into account, as its convenience is one of the requirements for a usable biometric system. Finally, extending databases with more conditions and user backgrounds could be beneficial to the research community, as it could lead to improving all the stages to make the modality more universal, i.e., including users with heart diseases or collecting data with cheaper, more usable devices.

Author Contributions: Conceptualization, P.T.-M. (corresponding author) and R.S.-R.; methodology, P.T.-M.; software, P.T.-M.; validation, P.T.-M.; formal analysis, P.T.-M.; investigation, P.T.-M.; resources, R.S.-R.; writing—original draft preparation, P.T.-M.; writing—review and editing, P.T.-M.; supervision, R.S.-R.; project administration, R.S.-R. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset employed in this study is not public.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rosenblatt, F. *The Perceptron, a Perceiving and Recognizing Automaton Project Para*; Cornell Aeronautical Laboratory: Buffalo, NY, USA, 1957.
2. Wason, R. Deep learning: Evolution and expansion. *Cogn. Syst. Res.* **2018**, *52*, 701–708. [[CrossRef](#)]
3. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [[CrossRef](#)] [[PubMed](#)]
4. Goldberger, A.; Goldberger, Z.; Shvilkin, A. *Clinical Electrocardiography: A Simplified Approach E-Book*; Elsevier Health Sciences: Amsterdam, The Netherlands, 2017.
5. Hoekema, R.; Uijen, G.J.H.; van Oosterom, A. Geometrical aspects of the interindividual variability of multilead ECG recordings. *IEEE Trans. Biomed. Eng.* **2001**, *48*, 551–559. [[CrossRef](#)] [[PubMed](#)]
6. Schijvenaars, B.J.; van Herpen, G.; Kors, J.A. Intraindividual variability in electrocardiograms. *J. Electrocardiol.* **2008**, *41*, 190–196. [[CrossRef](#)]
7. Jekova, I.; Krasteva, V.; Leber, R.; Schmid, R.; Twerenbold, R.; Müller, C.; Reichlin, T.; Abächerli, R. Intersubject variability and intrasubject reproducibility of 12-lead ECG metrics: Implications for human verification. *J. Electrocardiol.* **2016**, *49*, 784–789. [[CrossRef](#)]

8. Biel, L.; Pettersson, O.; Philipson, L.; Wide, P. ECG analysis: A new approach in human identification. *IEEE Trans. Instrum. Meas.* **2001**, *50*, 808–812. [\[CrossRef\]](#)
9. Goldberger, A.L.; Amaral, L.A.N.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* **2000**, *101*, E215–E220. [\[CrossRef\]](#)
10. Ribeiro, A.H.; Ribeiro, M.H.; Paixão, G.M.; Oliveira, D.M.; Gomes, P.R.; Canazart, J.A.; Ferreira, M.P.; Andersson, C.R.; Macfarlane, P.W.; Wagner, M.; et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nat. Commun.* **2020**, *11*, 1–9. [\[CrossRef\]](#)
11. Nemirko, A.P.; Lugovaya, T.S. *Biometric Human Identification Based on Electrocardiogram*; MAKS Press: Moscow, Russia, 2005; pp. 387–390.
12. Merone, M.; Soda, P.; Sansone, M.; Sansone, C. ECG databases for biometric systems: A systematic review. *Expert Syst. Appl.* **2017**, *67*, 189–202. [\[CrossRef\]](#)
13. Wang, Z.; Zhang, Y. Research on ECG Biometric in Cardiac Irregularity Conditions. In Proceedings of the 2014 International Conference on Medical Biometrics, Shenzhen, China, 30 May–1 June 2014; pp. 157–163. [\[CrossRef\]](#)
14. Odinaka, I.; Lai, P.H.; Kaplan, A.D.; O’Sullivan, J.A.; Sirevaag, E.J.; Kristjansson, S.D.; Sheffield, A.K.; Rohrbaugh, J.W. ECG biometrics: A robust short-time frequency analysis. In Proceedings of the 2010 IEEE International Workshop on Information Forensics and Security, Seattle, WA, USA, 12–15 December 2010; pp. 1–6. [\[CrossRef\]](#)
15. Jekova, I.; Krasteva, V.; Schmid, R. Human Identification by Cross-Correlation and Pattern Matching of Personalized Heartbeat: Influence of ECG Leads and Reference Database Size. *Sensors* **2018**, *18*, 372. [\[CrossRef\]](#)
16. Choudhary, T.; Manikandan, M.S. A novel unified framework for noise-robust ECG-based biometric authentication. In Proceedings of the 2nd International Conference on Signal Processing and Integrated Networks, Noida, India, 19–20 February 2015; pp. 186–191. [\[CrossRef\]](#)
17. Arsene, C.T.; Hankins, R.; Yin, H. Deep learning models for denoising ECG signals. In Proceedings of the European Signal Processing Conference, A Coruna, Spain, 2–6 September 2019. [\[CrossRef\]](#)
18. Srivastava, A.; Yadav, K.T.; Tiwari, R.; Venkateswaran, K. A Brief Study on Noise Reduction Approaches Used in Electrocardiogram. In Proceedings of the Second International Conference on Inventive Research in Computing Applications, Coimbatore, India, 15–17 July 2020; pp. 1023–1027.
19. Chamatidis, I.; Katsika, A.; Spathoulas, G. Using deep learning neural networks for ECG based authentication. In Proceedings of the 2017 International Carnahan Conference on Security Technology (ICCST), Madrid, Spain, 23–26 October 2017; pp. 1–6. [\[CrossRef\]](#)
20. Zhao, H.H.; Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. *Granul. Comput.* **2020**, *5*, 411–418. [\[CrossRef\]](#)
21. Mustaqeem; Kwon, S. MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach. *Expert Syst. Appl.* **2021**, *167*, 114177. [\[CrossRef\]](#)
22. Shewalkar, A.; Nyavanandi, D.; Ludwig, S.A. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235–245. [\[CrossRef\]](#)
23. Mekruksavanich, S.; Jitpattanakul, A. Biometric User Identification Based on Human Activity Recognition Using Wearable Sensors: An Experiment Using Deep Learning Models. *Electronics* **2021**, *10*, 308. [\[CrossRef\]](#)
24. Tan, R.; Perkowski, M. Toward Improving Electrocardiogram (ECG) Biometric Verification using Mobile Sensors: A Two-Stage Classifier Approach. *Sensors* **2017**, *17*, 410. [\[CrossRef\]](#)
25. Islam, M.S.; Alajlan, N. Biometric template extraction from a heartbeat signal captured from fingers. *Multimedia Tools Appl.* **2017**, *76*, 12709–12733. [\[CrossRef\]](#)
26. Kim, J.; Sung, D.; Koh, M.J.; Kim, J.; Park, K.S. Electrocardiogram authentication method robust to dynamic morphological conditions. *IET Biom.* **2019**, *8*, 401–410. [\[CrossRef\]](#)
27. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [\[CrossRef\]](#)
28. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
29. International Organization for Standardization. *ISO/IEC JTC 1/SC 37: Information Technology—Biometric Performance Testing and Reporting—Part 1: Principles and Framework (Standard No. 19795)*; International Organization for Standardization: Geneva, Switzerland, 2006; p. 56.
30. Foley, P.W.X.; Patel, K.; Irwin, N.; Sanderson, J.E.; Frenneaux, M.P.; Smith, R.E.A.; Stegemann, B.; Leyva, F. Cardiac resynchronisation therapy in patients with heart failure and a normal QRS duration: The RESPOND study. *Heart* **2011**, *97*, 1041–1047. [\[CrossRef\]](#)
31. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: Berlin/Heidelberg, Germany, 2006.
32. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
33. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2017**, arXiv:1609.04747.
34. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.

-
35. Jain, A.K.; Ross, A.; Prabhakar, S. An Introduction to Biometric Recognition. *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)* **2004**, *14*, 1–29. [[CrossRef](#)]
 36. Tirado-Martin, P.; Liu-Jimenez, J.; Sanchez-Casanova, J.; Sanchez-Reillo, R. QRS Differentiation to Improve ECG Biometrics under Different Physical Scenarios Using Multilayer Perceptron. *Appl. Sci.* **2020**, *10*, 6869. [[CrossRef](#)]