Differential Beat Accuracy for ECG Family Classification Using Machine Learning

ALBA VADILLO-VALDERRAMA, REBECA GOYA-ESTEBAN, RAÚL P. CAULIER-CISTerna, ARCADI GARCÍA-ALBEROLA, AND JOSÉ LUIS ROJO-ÁLVAREZ (Senior Member, IEEE)

1Department of Signal Theory and Communications and Telematics Systems and Computation, Rey Juan Carlos University, Fuenlabrada, 28943 Madrid, Spain
2Centro de Imagen Biomédica, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile
3Rielo Institute for Integral Development, New York, NY 11102, USA
4Cardiology Service, Arrhythmia Unit, Hospital General Universitario Virgen de la Arrixaca, El Palmar, 30120 Murcia, Spain
5DilemmaLab Ldt., Fuenlabrada 28917, Spain

Corresponding author: Rebeca Goya-Esteban (rebeca.goyaesteban@urjc.es)

This work was supported by the Science and Innovation Ministry Grants meHeart-RisBi and Beyond under Grant PID2019-104356RB-C42/AEI/10.13039/501100011033 and Grant PID2019-106623RB-C41/AEI/10.13039/501100011033. The work of Raúl P. Caulier-Cisterna was supported by ANID Fondecyt Postdoctorado 2021 under Grant 3210305.

ABSTRACT Holter systems record the electrocardiogram (ECG), which is used to identify beat families according to their origin and severity. Many systems have been proposed using signal conditioning and machine learning (ML) classification algorithms for beat family recognition. However, the design stage of these systems does not always consider the impact that tuning the intermediate blocks has on the beat family classification and the overall accuracy. We propose to use a new index based on the confusion matrices and bootstrap resampling to summarize the global performance for all family beats, so-called differential beat accuracy (DBA), which is obtained as the total number of beats correctly classified in each class minus the total number of beats incorrectly classified. We addressed the sensitivity of the different subblocks when creating a simple beat family classifier consisting of signal preprocessing blocks and a simple k-Nearest Neighbors classifier. The MIT-BIH Arrhythmia database was used for this purpose, following existing literature on the field. We benchmarked two implementations, one for biclass classification (supraventricular vs. non-supraventricular origin) and another for multiclass beat labeling. The usual preprocessing stages were scrutinized with the DBA to evaluate their impact on the quality of the complete ML system, such as signal detrending and filtering, beat balancing, or inter-beat distance. With the support of the DBA, our methodology was able to detect significant differences in terms of some of the options in the algorithm design. For instance, balancing the number of beats in each class for training significantly improved the classification accuracy of the minority classes at 3.22% for the multiclass dataset but not for the biclass dataset. Also, accuracy improved significantly by about 6% for the biclass regrouping without data normalization, whereas overall accuracy improved significantly by about 7% for the multiclass regrouping with data normalization. In addition, the analysis of the statistical dispersion of confusion matrices showed that this database should be considered with caution when training ML-based family classifiers. We can conclude that the proposed DBA can provide us with statistically principled criteria for designing ML-based classifiers and reducing their bias in strongly unbalanced beat family datasets.

INDEX TERMS Machine learning, ECG morphology, MIT-BIH database, family beats, differential confusion matrix, differential beat accuracy.

I. INTRODUCTION Cardiovascular diseases are one of the leading causes of death worldwide, directly concerning public health. The most common method to diagnose and treat heart disease is the electrocardiogram (ECG), which represents the electrical activity of the heart [1] and an informative non-invasive medical test [2], [3]. Short-time ECG recordings are commonly used in a daily clinical cardiology routine due to their easy acquisition. In contrast, in some cases, a longer follow-up is
necesary due to transient pathological events that may not be detected in short recordings [4], [5]. On the other hand, Holter systems are commonly used as a non-invasive tool in ambulatory monitoring, consisting of a portable recording device with attached cutaneous electrodes on the chest wall [6] that yields better chances to identify arrhythmias or cardiac abnormalities in patients. Holter monitors can use 2, 3, or 12 electrodes which can record the ECG signal during periods of 24 or 48 hours, and recent monitors can offer up to weeks of monitoring [7]. The data are stored in the device using digital media, analyzed with software created by a technologist, and subsequently edited and reported by the physician. A usual automatic processing stage of Holter analysis consists of the identification of different morphologies for the beats in a patient recording, which are usually known as beat families, and these families are subsequently used to support the cardiologist on the higher-level rhythm analysis or cardiac disorder identification for that patient [8], [9], [10]. The process of identifying these beat families, also known as beat labeling, needs to be semisupervised in clinical practice, which means that the software in current Holter systems is capable of making an initial grouping based on some internal criteria, and it is subsequently readjusted by the clinician [11], [12].

In recent decades, algorithms and criteria have been dedicated to automating the heartbeat classification stage. These are primarily based on rules about time intervals and characteristic voltage levels in each beat. Still, this approach has limited precision due to the sensitivity of the processing system to different parameter thresholds required in other patients [13], [14], [15]. Cardiologists reviewing the family grouping can change those thresholds when necessary in each patient and discard those families corresponding with artifacts or irrelevant information. However, this is a highly time-consuming task for the specialist, and the complete process can take between 10 to 45 minutes per patient [16]. Moreover, the beat family detection stage depends on the previous ECG-preprocessing stages, which generally include different signal filters and some segmentation steps. Still, in general, little attention has been paid to the sensitivity of the family beat detection and to the design parameters of the preprocessing stages [17].

In the literature, we can find excellent reviews of applications of machine learning (ML) algorithms in medicine in general [18], [19], and in cardiology in particular, [20], [21]. In this setting, many methods have been proposed during the last years for beat classification in ECG recordings, which use artificial intelligence and ML techniques [1], [22], [23]. Despite the interest in ML-based solutions for beat family identification being expected to grow in the forthcoming years, few commercial Holter systems currently include their use, despite the amount of academic literature devoted to its research. Some preprocessing stages and their design parameters are expected to impact the final performance of ML schemes significantly. Still, to our best knowledge, standard ECG preprocessing stages are often implemented without accounting for the sensitivity of ML algorithms to them. The sensitivity analysis of ML generalization capabilities in ECG preprocessing design could notably, contribute to support the use of these technologies in next-generation Holter systems.

Considering the preceding points, we propose using a new index based on confusion matrices and bootstrap resampling to summarize the global performance for all family beats, the so-called Differential Beat Accuracy (DBA). The usual preprocessing stages, such as signal detrending and filtering, beat balancing tuning, or inter-beat distances, are scrutinized to evaluate their impact on the quality of the input vectors for the complete ML system. In summary, we propose a methodology to assess the effect of the signal preprocessing and machine learning options on the final beat families classification results, with a focus on the generalization capabilities of the system. The said methodology can support the design and performance evaluation of machine learning systems in terms of clear cut-off tests which are non-parametric. Specifically, they are based on bootstrap resampling for scrutinizing the differences between two different processing schemes.

Accordingly, a note from this point on is that our focus is not to obtain the best ML system for this application but to determine the generalization capabilities of ML algorithms working on databases for Holter benchmarking. Several simplifications need to be made to fulfill this aim. First, we worked with the raw beat waveform and applied different preprocessing techniques such as filtering, baseline cancellation, and segmentation. Second, for the analysis above, we primarily work with a widely used (yet simple) ML algorithm, namely, the k nearest neighbors (KNN) algorithm for classification [24], [25], [26]. Others have overcome this algorithm in different studies [27], [28], [29]. Still, its performance is enough to analyze its generalization capabilities on various conditions in the scope of the present paper. Third, the MIT-BIH database has been considered in our experiments, as it has been widely used to design ML algorithms in the literature, which allows us to pay special attention to the unbalance in the beat labels in this database.

II. BACKGROUND

Cardiac signals have been used in a variety of applications of ML algorithms, including myocardial infarction diagnosis, arrhythmia discrimination, or hypertrophy detection [30], [31]. Other tasks related with cardiac signals have been addressed with this emerging technology, such as the heart rate variability analysis for classification of diabetic and healthy subjects, basic processing stages such as QRS-complex detection, or detailed wave delineation [32], [33]. In [34], an algorithm was developed which exceeded the performance of cardiologists in detecting a wide range of heart arrhythmias from ECGs recorded with a single-lead wearable monitor, using a proprietary dataset of 91,232 single-lead ECGs from 53,549 unique patients with 14 classes. This study can be pointed as an excellent reference work in the field. In the context of cardiac arrest, traditional ML architectures such as random forest or recursive least squares [35], [36],
The use of ML paradigms, specifically as a classification problem formulation, aims to improve the overall accuracy of the beat family detection system. This task needs to know the clinical labels of every beat in the database used for learning, which is a difficult requirement. Many authors [22], [40] have tried to overcome this need by using broadly available databases for technical benchmarking of Holter systems. A well-known example is the MIT-BIH Arrhythmia Database [41], a publicly available set of ECGs with detailed annotations and clinical information from different patients that has supported for years the research and system benchmarking in the ECG processing field. Methods of sparse dictionary learning, which aim to obtain a combination of basic elements that represent the input data sparsely, have several applications in data decomposition, compressed sensing and signal recovery [42], [43]. This approach has been applied to the fields of image denoising and classification, video and audio processing [44], [45], as well as to medical signal analysis, as electroencephalography (EEG), ECG, magnetic resonance imaging (MRI), functional MRI, continuous glucose monitors [46], and ultrasound computer tomography, where different assumptions are used to analyze each signal. ECG signals can also be factorized in coefficients in order to get a dictionary and to use it to extract features to be used in ML-based beat classification schemes [47], [48].

Moreover, contributions of DL to ECG have grown immensely in the last 5 years, most of them on classification applications, and especially in arrhythmia detection and classification, but also for diagnosing atrial fibrillation during normal sinus rhythm, cardiac dysfunction, sleep apnea, or hypertension, as well as with biometric purposes. In [49], an overall view of DL for biomedical applications can be found, and thorough reviews of processing-based DL in ECG are provided in [50], [51], [52], [53], and [54]. Also, in [55], a deep review of DL in ECG is carried from a clinical standpoint. In general, these authors highlight the advantages of processing raw data over the traditional ML based on feature extraction, which yields better performance, though with several limitations. Foremost, a fair comparison among the different approaches becomes very difficult due to the diversity in ECG input data and preprocessing blocks. Another challenge to be faced is generalization. Most of the reported studies use a single dataset, mainly from a public database (Physionet), resulting in potential bias due to the small size. Although it may apparently seem a large and diverse amount of data in terms of beat availability, it can fail in terms of number of patients (intra-patient redundancy). The imbalance of classes is also a concern. The lack of explainability or interpretability is also referred to as a problem because it provides a black-box model (straight from waveform to event detection) that humans cannot understand, so this drawback should also be solved to achieve the trust of clinicians and to allow the transference of ML and DL algorithms into the clinical practice [50], [55]. In the present work we certainly devoted our best effort to provide clear cut-off tests for model benchmarking, while paying attention to the imbalance and to the generalization capabilities. Furthermore, in recent years, new trends have emerged to explain ML models in order to understand the process of each of the systems delivered from this area. This is highly important for many regulated sectors, such as the biomedical sector. A widespread used reference for these new trends is the method known as the Local Interpretable Model-Agnostic Explanations (LIME) [56], which explains the ML predictions of many classifiers in an interpretable way, and this method has been used when dealing with ECG classifications [57]. Another helpful technique used in the recent years is t-SNE, which is used for for high-dimensional data visualization in the area of ML and DL. [58]. In the present work, we use both LIME and t-SNE to interpret our results.

Note that the focus of this work is not to deliver a new classification method, but rather we aim to create a statistically-principled method capable of supporting the design of signal preprocessing stages and ML tuning steps when creating a beat classification system. Therefore, several of these relevant aspects of the background are not addressed here, but rather we use a simple classifier (the KNN classifier) in order to prove the usefulness of the proposed method and the use of the DBA index.
III. METHODS

In this section, we detail the different digital processing stages of beat family systems, including beat regrouping, preprocessing, and classification. The proposed and used bootstrap-resampling based statistical descriptions are stated. These statistical indices allow us to make principled and informed decisions on the system design according to the overall ML system performance, showing both a fine-grain evaluation view (with the differential confusion matrix) and an overall evaluation view (with the DBA). Figure 1 presents a summary scheme of the methodology used in this work, and each of the steps is explained below.

A. SIGNAL PREPROCESSING

Cardiac signals often present noise, which can be due to interference with the loss of electrode contact with the skin or to other physiological origins. This noise distorts the signals, thus hindering their analysis and their posterior study [59]. In order to obtain advantageous results in the analysis of ECG signals, it is important to establish an adequate digital preprocessing pipeline, so that we first focus on cleaning the noise present in the signal while preserving its morphology. Every ECG contained in a database consists of a collection of consecutive beats, and it is denoted generally as \( x[n] \), where \( n \) is the number of sample and \( x[n] = x(nT_s) \), where \( T_s \) is the sampling period of the continuous time recording \( x(t) \).

The baseline is an undesired external interference of low frequency activity on the ECG, and it is usually the result of various sources of noise such as breathing, body movements, or poor contact of the electrodes with the skin [59]. In the present work, the baseline cancellation (BLC) was performed with the combination of a sliding window, a median filter for the samples of this window, and spline interpolation on the nodes in the window central positions [60], which can be denoted by using operator \( \Gamma_{BLC} \) to modify the discrete-time ECG signal, as follows,

\[
x_{BLC}[n] = \Gamma_{BLC}(x[n])
\]  

(1)

After BLC, high frequency noise is usually filtered out and for this purpose we applied a low-pass filter using operator \( \Gamma_{FIL} \), thus allowing us to obtain the processed signal denoted as \( x_{FIL}[n] \), as follows

\[
x_{FIL}[n] = \Gamma_{FIL}(x_{BLC}[n], \theta^1_{FIL}, \theta^2_{FIL})
\]  

(2)

where \( \theta^1_{FIL} \) and \( \theta^2_{FIL} \) are free parameters indicating the order of the filter and the cut-off frequency.

As other databases, the MIT-BIH database provides the positions of the QRS complexes of each beat for all the patients, hence we can work with R peaks extracted from \( x[n] \) by finding the sample with the maximum value in a window value around each beat, and its corresponding value of \( n \) which indicates the position of the sample. Segmentation can be performed based on these peaks and a time interval is considered before (denoted as \( t_1 \)) and after (denoted as \( t_2 \)) the peaks. In summary, in the segmentation process we obtain the beats of each patient and each beat is composed of a set of samples, hence providing with information about the ECG signal in terms of the beat morphology. The values of the samples of these beats can be used as features when classifying them with subsequent ML systems.

We used here a convenient representation of the signals to support the analysis, which is the so-called M-mode. This representation is just a three-dimensional plot of all the segmented beats of one patient in which the \( x \) axis represents the number of beat, the \( y \) axis represents the relative time of each beat, and the \( z \) axis represents the amplitude of each beat [61], as we can see in the example of Figure 2. This representation is used to analyze the effect of filtering in specific and representative cases, and it can be also useful to visualize quickly and effectively if there is any anomaly in the beats from a given patient or to see the noise present in the segmented signal.

B. KNN ALGORITHM

The ML classification method implemented in this work is the KNN, which is a simple supervised classification algorithm in which the database is divided in two subgroups, the training data and the test data. The training data, which helps to generate the model, is given by a matrix that contains the segmented beats of the patients selected for this subset, defined as \( X_{tr} \) where each row corresponds to one beat and the columns correspond to beat samples. The test data, denoted in matrix form as \( X_{ts} \), allows us to measure the generalization capacity of that model [40] and it is defined as \( X_{tr} \) but containing the beats from the patients selected for this subset instead.

Both groups are labeled and these labels are denoted in vector form as \( y_{tr} \) and \( y_{ts} \), both being column vectors. Therefore, the ECG samples were used as the input features in the KNN algorithm, and no further feature engineering was considered here, for simplicity.

The algorithm classifies a new instance \( x_{ts} \) from \( X_{ts} \) by calculating the distances between this instance and all the instances in \( X_{tr} \). Afterwards the \( k \) closest distances are selected, and the new instance is assigned to the majority class among its \( k \) nearest neighbors in \( X_{tr} \). The used distance metric is often the Euclidean distance [62], denoted as follows,

\[
d_e(x_{tr}, x_{ts}) = \sqrt{\sum_{i=1}^{N_t} (x_{ts,i} - x_{tr,i})^2}
\]  

(3)

where \( N_t \) represents the number of time samples in a segmented beat.

In this work we also tested other distance metrics, namely, the Manhattan distance and the correlation distance. The Manhattan distance, \( d_m \), is denoted as follows,

\[
d_m(x_{tr}, x_{ts}) = \sum_{i=1}^{N_t} |x_{ts,i} - x_{tr,i}|
\]  

(4)
FIGURE 2. Representation of different segments of ECGs. (a) Two example segments of ECGs in the used database. The y axis represents the amplitude of the signal in mV and the x axis represents the time in seconds. The upper (lower) panel shows a recording with normal beats (with some beat anomalies). (b) M-mode representation of available beats in an example patient.

TABLE 1. Examples of imbalance in the number of labels (multiclass dataset).

| Patient | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|---------|---------|---------|---------|---------|---------|
| Patient 1 | 2239    | 33      | 1       | 0       | 0       |
| Patient 2 | 1860    | 3       | 0       | 0       | 2       |
| Patient 3 | 99      | 0       | 4       | 0       | 56      |
| Patient 4 | 2082    | 0       | 0       | 0       | 0       |

and the Correlation distance, \(d_c\), is denoted as follows,

\[
\rho(x_{tr}, x_{ts}) = \frac{\sum_{i=1}^{N_t} (x_{ts,i} - \bar{x}_{ts})(x_{tr,i} - \bar{x}_{tr})}{\sqrt{\sum_{i=1}^{N_t} (x_{ts,i} - \bar{x}_{ts})^2} \sqrt{\sum_{i=1}^{N_t} (x_{tr,i} - \bar{x}_{tr})^2}}
\]

\[
d_c(x_{tr}, x_{ts}) = 1 - \rho(x_{tr}, x_{ts})
\]

where the average of each beat vector is denoted with the bar. We want to stress that we used the KNN classifier because of its simplicity, and although it may not be the most advanced classification technique, it serves us for evaluating the impact of the different preprocessing stages on an overall classification system, according to the scope of this paper.

C. BALANCING AND NORMALIZATION

The used databases often present a complicated problem, namely, they are strongly unbalanced even when regrouping the existing subclasses. This means that the total number of beats per class is very uneven and it increases the complexity when classifying with ML systems. As we can see in some examples of the MIT-BIH database shown in Table 1, where the columns indicate the number of beats belonging to the different regrouped classes for each example patient, we can see that most of the beats of every patient belong to the first class (sinus rhythm) and there are many classes in many patients that have no beats.

In order to have a more balanced database we separate selectively some patients into \(X_{tr}\), and some other patients into \(X_{ts}\), the rest of the patients being assigned randomly. This selection is made based on the number of beats in each class for each patient, so that both the test set and the train set contain beats from every class. In addition, different sizes of the database were tested by reducing the maximum number of beats in every class for every patient, and this value is called \(N_m\). Recall here that all the beats from one patient go either to the training set or to the test set, resembling a real scenario where beats from a new patient have not been used to train the ML algorithms, and thus avoiding the risk of reporting non-realistic classification accuracy values.

Statistical normalization methods are another preprocessing option that can be considered. These techniques scale the values of our data without distorting differences in the range of values, in this work we normalized the data to zero mean and unit standard deviation for each beat as follows,

\[
x_{tr}' = \frac{1}{\sigma}(x_{tr} - \mu)
\]

where \(\mu\) and \(\sigma\) stand for the mean and the standard deviation of that given beat, and it is similarly normalized for the test beats.

IV. BOOTSTRAP AND CONFUSION INDICES

In this section, a Bootstrap resampling scheme is presented for its use on the evaluation of the differences among signal preprocessing and ML tuning options. The objective of this procedure is to characterize the statistical sampling variability for drawing inferences about the population based on the results on our dataset, while accounting for the impact of a preprocessing design decision on the overall performance, and also while taking into account the inter- and intra-class performance, especially in cases of multiple and strongly unbalanced labels in the test set.
We can study sampling variability by artificially sampling with replacement the originally available population, so that this new but repeated population can be the dataset from which we seek to draw inferences. Since the dataset is itself a sample of the whole population, we are taking a sample from the sample, i.e., we are resampling. This does not provide more information from the population, but it rather provides us with the quantification of sampling variability for drawing inferences about the population based on our data [63]. The resampling method can be used to compute confidence intervals (CI) of many types of statistics and to perform cut-off hypothesis tests.

In the present problem, we need to decide whether the performance differences between two preprocessing options are statistically significant in terms of some given and selected performance statistics. Our statistical hypothesis test contrasts the null hypothesis \( H_0 \) that preprocessing options yield the same performance, against the alternative hypothesis \( H_1 \) that they yield different performance on the overall ML system, this is,

\[
H_0 : \Delta u = 0 \\
H_1 : \Delta u \neq 0,
\]

where \( u_1 \) and \( u_2 \) denote the performance statistic obtained for each set of preprocessing options, and \( \Delta u = u_1 - u_2 \) is any of the differential statistics used here for hypothesis testing.

In order to approximate the probability density function (pdf) of \( u_1, u_2, \) and subsequently of \( \Delta u \), we use the well-known plug-in principle. In general, let \( Z = \{z_j, j = 1, \ldots, L\} \) be a set of \( L \) measures, and let \( u \) be a statistical magnitude estimated by using an operator \( O \) on the observed set, i.e., \( u = O(Z) \). Since the actual \( f_2(Z) \) is likely to be unknown and only a finite number of samples are available, and the operator \( O \) can be complex to obtain analytically, it turns out that \( f_2(u) \) will be often impractical to compute. Alternatively, we can approximate \( f_2(Z) \) by its plug-in empirical distribution, \( f_2(Z) \). We build sets \( Z^*(b) \) (so-called resamples from \( Z \)), by sampling with replacement up to \( L \) elements of \( Z \). Now, a replication of the statistic \( u \) is obtained as \( u^*(b) = O(Z^*(b)) \), and it represents an estimate of this statistic. By repeating the resampling procedure for \( b = 1, \ldots, B \) times, an estimated pdf for our performance statistic is given by

\[
f_\hat{u}(u) = \frac{1}{B} \sum_{b=1}^{B} \delta(u - u^*(b))
\]

An estimation of the CI for \( \Delta u \) can be similarly and readily obtained from ordered statistics in \( \Delta u^*(b) \) resamples [64]. The differences between the two methods are considered as statistically relevant in terms of statistic \( u \) when the 95% IC of \( \Delta u \) does not overlap the zero value.

In the present problem, \( Z \) stands for the beat classification labels and \( u \) stands for the confusion matrix \( (M) \) generated with the two possible hypotheses \( (M_1 \) and \( M_2 \), respectively), where

\[
M = [m_{u,v}]
\]

and element \( m_{u,v} \) denotes the number of observations known to be in class \( u \) but predicted to be in class \( v \). We start from two vectors containing the beat classification labels for the two preprocessing options, named \( Z_1 \) and \( Z_2 \). In each resampling iteration \( b \) these vectors are resampled, and also \( y \) is resampled. In each iteration \( b \), confusion matrices \( M_1 \) and \( M_2 \) are obtained, corresponding to the two resampled vectors, and in addition \( \Delta M = M_1 - M_2 \) is obtained. At the end of the \( B \) replications, an estimation of the pdf of each element of the \( \Delta M \) matrix is obtained to determine statistically significant differences. Then, the 95% IC is calculated for each element in \( \Delta M \), and if it does not contain zero, the differences between both preprocessing methods are considered as statistically significant, and the previously obtained matrix \( \Delta M \) is...
multiplied by one (zero) those values that are (that are not) significant according to their IC, and we obtain \( \Delta M_2 \).

Finally, we also define the DBA, denoted as \( S_d \), as the total number of beats correctly classified in each class minus the total number of beats incorrectly classified, when moving from the set of preprocessing options 1 to the set of preprocessing options 2. The statistic is hence calculated as follows:

\[
S_d = 2 \sum_{u=1}^{N_c} m_{u,u} - \sum_{u=1}^{N_c} \sum_{v=1}^{N_c} m_{u,v}
\]

where \( N_c \) stands for the number of beat classes. The histogram of this statistic, \( S_d \), can be plotted and analyzed, in such a way that if the CI overlaps the zero, the effect of the difference in the preprocessing options (1 and 2) is considered as non-significant, and if it does not overlap zero, the difference between said preprocessing options is significant from an overall point of view. When the differential beat accuracy \( S_d \) is significant, it can be positive or negative, noting that when it is positive (negative) this means that the first (second) preprocessing option is better than the second (first) one as shown Figure 3 (a). Parameter \( S_u \) is also multiplied by one (zero) those values that are (that are not) significant according to their CI, obtaining a narrower histogram for \( S_b \) as shown in Figure 3(b). This results in a reduction of the standard error of the estimators, and an increase in the statistical power of the tests.

V. EXPERIMENTS AND RESULTS

In this section, we first describe a set of experiments conducted with computationally obtained surrogate beats. These experiments show the performance of the proposed methodology in a known-solution case. Following, we describe the real database used in this work, proceeding from Physionet [41] and previously used in many related works. Then, we scrutinize the implementations of different processing options, and these options are compared and analyzed using the proposed Bootstrap statistical differential tests and confusion matrices. The different options are: the data normalization; selecting the maximum number of beats per patient (\( N_m \)) to feed the classifier; the distance metric implemented in the KNN algorithm, as well as the number of neighbors (\( k \)); the baseline cancellation and the low pass filtering. We tested all these preprocessing options both for the multiclass regrouping and for the biclass regrouping.

A. SYNTHETIC DATABASE

To show the usefulness of the proposed methodology in a known-solution case, different data sets composed of surrogate beats were generated with software developed in Matlab. This software allows us to simulate heartbeats from different families following a set of rules given by cardiologists, namely, the duration of the QRS complex and PQ and ST intervals and the amplitude and symmetry of the heartbeats waves. Figure 4 shows some examples of synthetic heartbeats, including normal sinus rhythm (SR) beats in Fig. 4 (a), supra ventricular (SV) beats in Fig. 4 (b), ventricular (VT) beats in Fig. 4 (c), and noise in Fig. 4 (d).

First, we generate 1500 beats for each class and divide them randomly into three sets of 500 beats, namely, train, validation, and test sets. Secondly, we find the \( k \) value (scrutinizing from 1 to 45) that minimizes the classification error in the validation set, and it turns out to be \( k = 13 \). However, we use the proposed bootstrap statistical procedure to find the \( k \)-value where the classification error decrease is statistically significant since a large \( k \)-value is prone to overfitting. We start comparing increasing \( k \)-values from 3. Figure 5 shows the histogram of \( S_b \) a) comparing \( k = 15 \) and \( k = 3 \), and b) comparing \( k = 13 \) and \( k = 9 \). The histogram in (a) shows that \( k = 13 \) significantly outperforms \( k = 3 \). It is significant because the 95% CI of \( S_b \) does not overlap zero and is better for the first option (\( k = 13 \)) because the 95% CI of \( S_b \) is composed of positive values. The histogram in (b) shows no statistical difference between the results with \( k = 13 \) and \( k = 9 \).

We then used LIME to identify the input characteristics, in our case, the signal samples, which the classification algorithm considers to classify the cases. Figure 6 shows one example for each class, namely, (a) SR beat, (b) SV beat, (c) TV beat, and (d) noise. For each example, the top panel shows the beat, and the middle panel shows the signal samples that LIME determines that are used to classify the example. The bottom panel shows the same samples weighed by their importance in the classification.

Also, t-SNE is used to find a low-dimensional representation of the data. Figure 7 (a) shows how the different classes of surrogate data are distributed in a low dimensional space. It can be seen that the main overlapping occurs between SR and SV families. Figure 7(b) shows an example of a misclassified beat labeled as SR but classified as SV. LIME shows us that this is probably due to the low level of the p-wave and that it also uses information from the QRS-complex and the repolarization segment for the classification. Note that this example can be closer to the pattern in Fig 6(b) than the pattern in Fig 6(a). Also, some SV beats can be misclassified as RS because they do present some t-wave and the algorithm determines that they are more close to the pattern in Fig 6(a) than to the pattern in Fig 6(b).

For the next experiment, we generated two biclass datasets with balanced data (8000 SR beats and 8000 SV beats) and unbalanced data (8000 SR beats and 1000 SV beats). The experiment evaluated the effect of the unbalances in the data by comparing the performance of the KNN algorithm between both datasets. However, the confusion matrices showed that we were not at the local optimum in which the machine gives the majority class solution, and more, we obtained no statistically significant difference from both learning schemes, which shows that this unbalance is acceptable for building a KNN-based beat classifier in these conditions.

For the last experiment with synthetic data, we generated two multiclass datasets. One dataset comprises the same
FIGURE 4. M-mode representations depicting surrogated generated beats: (a) sinus rhythm (SR); (b) supraventricular beats (SV); (c) ventricular tachycardia beats (VT); and (d) noise.

FIGURE 5. Histogram of $S_b$ for the surrogate database. (a) comparing $k = 15$ and $k = 3$; (b) comparing $k = 13$ and $k = 9$.
	number of beats for each class, 8000 SR beats, 8000 SV beats, and 8000 VT beats. A second dataset included a different number of beats for each class, 8000 SR beats, 1000 SV beats, and 1000 VT beats. Figure 8 shows the histogram of $S_b$ with the comparison between both options, and it reflects that using the balanced dataset (first option) is significantly better than using the imbalanced dataset (second option). It is significant because the 95% CI of $S_b$ does not overlap zero, and it is better for the first option because the 95% CI of $S_b$ is composed of positive values. Table 2 shows the subtraction
of the confusion matrix of both options 2 (a), the confusion matrix of the first option $M_1$ in Table 2 (b) and the confusion matrix of the second option $M_2$ in Table 2 (c), note that the values of the diagonal of $\Delta M_2$ are positive and this indicates that the first preprocessing option is better than the second one.
B. DATA DESCRIPTION, BEATS, AND REGROUPING

Physionet is a website designed to promote and facilitate research in biomedical and physiological signals. In this paper, we used the MIT-BIH Arrhythmia Database [65], proceeding from its web [41]. This database has already been used in several works [8], [40], and it contains 48 half-hour extracts of two-channel ECG records obtained from 47 hospitalized patients, all of them sampled at 360 Hz. Figure 2 (a) shows two ECG fragments from two different patients. These records have been labeled by experts and are provided with their corresponding annotations according to different heart rhythms, which allows us to know how many classes there are and how many beats are in each class. There are originally a total of 19 different labels, namely, normal beat (N), left bundle branch block (L), right bundle branch block (R), bundle branch block beat (B), atrial premature beat (A), aberrated atrial premature beat (a), nodal premature beat (J), supraventricular premature or ectopic beat (atrial or nodal) (S), premature ventricular contraction (V), R-on-T premature ventricular contraction (r), fusion of normal and ventricular beat (F), atrial escape beat (e), nodal (junctional) escape beat (j), supraventricular escape beat (atrial or nodal) (n), ventricular escape beat (E), paced beat (f), fusion of paced and normal beat (f), unclassified beat (Q), and beat not classified during learning (?).

It can be seen that there is much detail in the label definition for the beats, which is not necessary for beat families classification in Holter systems, hence, it is usual in the literature to regroup these labels into some reduced set of types.
Two different criteria are used in the present work for label regrouping. The first one is multiclass labeling which classifies the beats into five types (N, S, V, F, U), as recommended by the standard from the Association for the Advancement of Medical Instrumentation (AAMI), and which is used in many works [22], [66], [67], it is shown in Table 3. However, these two previous label sets may lack clinical sense since they mix different criteria (QRS morphology, beat origin, rhythm). Therefore, the second one, a biclass regrouping, was stated by an expert cardiologist in our group according to its clinical usefulness, and this labeling mostly differences between the supraventricular and ventricular originated beats, as seen in Table 4. Figure 9 shows the histograms for the two types of labels.

As explained before, one should note that the digital processing pipeline in this problem is a hybrid, including signal preprocessing steps and ML steps. The first steps correspond to signal preprocessing (detrending, filtering, R-wave detection) and the subsequent ones to ML (beat segmentation, label assignment to the available beats for training and testing). Whereas label assignment to each segmented beat is at the end of the technical workflow, it is part of the description of the MIT-BIH database, which consists of a set of signals with time marks on their beats and with labels assigned by experts to those beats.

### C. MULTICLASS DATASET

#### 1) DATA NORMALIZATION

The first experiment consisted of checking how the data normalization impacts the final results of the ML classifier. This analysis was conducted by comparing the differential confusion matrix ($\Delta M_2$) and the differential beat accuracy ($S_b$) with (first preprocessing option) and without (second preprocessing option) using the normalization of the segmented beats provided as the ML input vectors. The remaining preprocessing parameters used in this initial configuration were: $N_m = 50$ as the maximum number of beats selected from each class per patient; the low-pass filter cut-off frequency was $\omega_{IL} = 60$ Hz, and the distance used in the KNN classification algorithm was the Euclidean one ($d_e$).

Figure 10 shows the histogram of $S_b$ for these two preprocessing options. Considering that $\Delta M_2$, as shown in Table 5 (a), is the subtraction of the confusion matrices; $M_1$, first option; $M_2$, second option.

| Classes          | Beats | Mean (Std) | MIT-BIH beat type |
|------------------|-------|------------|-------------------|
| Non-ectopic (N)  | 90.631| 1.888,1 (710.18)| N, L, R, e, j, j |
| Supra-ventricular ectopic beat (S) | 2.781 | 57.93,3 (206.30) | A, a, s, f |
| Ventricular ectopic beat (V) | 7.236 | 150.75 (242.48) | V, f |
| Fusion beat (F)  | 803   | 16.72 (73.99)  | F, q |
| Unknown (U)      | 8.043 | 167.56 (361.54) | /, f, q |

#### TABLE 5. Description of the biclass regrouping, where Beats stands for the total number of beats, and Mean (Std) stand for the mean (standard deviation).
option is better than the first one, and vice-versa. The histogram shows that all values within the 95% CI are positive, which indicates that the statistical difference between the two compared options is positive and that the first option gives a significantly higher number of correctly classified beats. This can also be analyzed with detail by looking at $\Delta M_2$. Therefore, we chose to normalize the data because a statistically significant gain turns into about 3600 better-classified beats. Furthermore, the observation of the elements in $\Delta M_2$ indicates that the first and fifth classes significantly increase their accuracy (positive diagonal elements) thanks to several confusion errors reducing (negative non-diagonal elements).

2) BALANCING THE TRAINING DATASET

The database is unbalanced for both regrouping schemes, as we have seen in Table 1, which shows the beats of some example patients for each of the classes. In this experiment, we tested different values of $N_m$, namely 50, 100, and using all the beats in all the training patients. The remaining preprocessing parameters were set as follows: $\theta_{FIL}^{2} = 60$ for low-pass filter cut-off frequency; data normalization as explained in the previous section; two neighbors and euclidean distance in the KNN algorithm.

We first compared $N_m = 50$ as the first preprocessing option using all the beats in each training patient as the second preprocessing option. The resulting histogram shows that the values in the 95% CI are negative, meaning that the second preprocessing option is better, and using all the beats without selecting them gives better overall performance than using a maximum of $N_m = 50$ beats per training patient. But if we analyze the differential confusion matrix, $\Delta M_2$, as shown in Table 6, we can see that the classes that improve the classification, the first and fifth class, match with the classes with more beats as shown Table 3. This does not mean that using the second preprocessing option improves the results on a per-class basis. Also, it does not mean that it is a better option than the first one, given that, in this case, the classes with more beats are classified better, which means that there could be overfitting.

3) DISTANCE METRIC IN THE KNN ALGORITHM

In this experiment, we tested how the different distance metrics implemented in the KNN algorithm affect its overall performance, considering the Euclidean distance ($d_e$), the Correlation distance ($d_c$), and the Manhattan distance ($d_m$). The remaining preprocessing parameters were set as follows:

4) NUMBER OF NEIGHBORS USED IN THE KNN ALGORITHM

In this experiment, we evaluated the effect of the number of neighbors used in the KNN algorithm, and for this experiment, we used: $N_m = 50$ as the maximum number of beats selected from each class per patient; low-pass filter cut-off frequency used was $\theta_{FIL}^{2} = 60$; and data normalization was included.

We first compared $d_e$ and $d_m$, and the histogram in Figure 11 (a) shows that using any of these two preprocessing options does not provide any difference at all. This is just because when the data are normalized, the distances are exactly the same due to their mathematical equation being coincident in this case, as expected. Secondly, we compared the $d_c$ with $d_m$. As we can see in Figure 11 (b), all $S_i$ values are negative, which means that the second option gives an advantage compared with the first option, thus $d_m$ was the preprocessing option selected over the other distances from now on. The differential confusion matrix showed that with the $d_m$ the first and the fifth class improve compared to the other distances.

### Table 6. Differential confusion matrix ($\Delta M_2$), obtained as the subtraction of the confusion matrices. The first (second) preprocessing option uses $N_m = 50$.

|        | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|--------|---------|---------|---------|---------|---------|
| Class 1| -1075   | 1263    | -276    | 100     | -12     |
| Class 2| 63      | 40      | -103    | 0       | 0       |
| Class 3| -235    | 61      | 85      | 61      | 28      |
| Class 4| -17     | 13      | 0       | 5       | 0       |
| Class 5| -36     | 25      | 0       | 63      | -59     |

### Table 7. Differential confusion matrix ($\Delta M_2$) obtained from the comparison of the first (second) preprocessing option, using $k = 4$ ($k = 10$) in the KNN algorithm.

|        | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|--------|---------|---------|---------|---------|---------|
| Class 1| -461    | 284     | 89      | 92      | 0       |
| Class 2| 31      | 16      | -45     | 0       | 0       |
| Class 3| 0       | 81      | 37      | -80     | -43     |
| Class 4| 10      | -9      | 7       | -8      | 0       |
| Class 5| 0       | 10      | 0       | 0       | -9      |
frequency $\theta_{FIL}^2 = 60$ Hz, and $d_m$ in the KNN classification algorithm.

Firstly, we compared using two neighbors ($k = 2$) with using four neighbors ($k = 4$). Figure 11 (c) shows the histogram with the comparison between both preprocessing options, which is not significant in this case. Secondly, we compared $k = 4$ with a higher number of neighbors of $k = 10$. This second comparison is shown in Figure 11 (d) and it reflects that when the number of neighbors increases, the differential accuracy improves. But if we analyze $\Delta M_2$, as Table 7 shows, it is similar to what we obtained in the previous subsection, because increasing the number of neighbors improves the classification for the first class, and it also does slightly for classes four and five, while classification for classes two and three worsens in a larger percentage. Therefore, using a high value for $k$ could lead to overfitting and decreased performance when trying to generalize the ML problem. We tested even and odd values for $k$ to avoid tie situations since the number of output classes is odd.

5) BASELINE CANCELLATION

In this experiment, we compared the baseline cancellation of the ECG signal with the baseline non-cancellation settings. The remaining preprocessing parameters were set as follows: $N_m = 50$, $\theta_{FIL}^2 = 60$, $k = 2$ and $d_m$ in the KNN classification algorithm.

As we can see in Figure 12, the histogram shows no significant difference for $S_b$. If we analyze $\Delta M_2$, as shown in Table 8, with cancellation the first and the fourth classes get worse, whereas the third and the fifth get better, and since the global improvement is in percentage higher the global worsening, we chose to include baseline cancellation. Figure 13 shows the effect of the baseline cancellation in every beat of one example patient.

6) LOW-PASS FILTERING

In this experiment, we compared the low-pass filter cut-off frequencies. The remaining preprocessing parameters were set as follows: $N_m = 50$, baseline cancellation, $k = 2$, and $d_m$ in the KNN classification algorithm.

Figure 14 (a) shows the comparison $\theta_{FIL}^2 = 60$ Hz as the first preprocessing option and $\theta_{FIL}^2 = 70$ Hz as the second preprocessing option. In this case, all the values are positive, and it does not overlap zero, which means that using $\theta_{FIL}^2 = 60$ Hz as frequency is better than using $\theta_{FIL}^2 = 70$ Hz. We also compared $\theta_{FIL}^2 = 60$ Hz and $\theta_{FIL}^2 = 50$ Hz, as shown in Figure 14 (b), and in this case, all the values overlap the value zero, that means that using $\theta_{FIL}^2 = 60$ is a similar option than using $\theta_{FIL}^2 = 50$ Hz when filtering. Finally, this cut-off frequency was compared with a lower value, $\theta_{FIL}^2 = 40$ Hz. As we can see in Figure 14 (c), the comparison between $\theta_{FIL}^2 = 50$ Hz as the first preprocessing option and

### Table 8. Differential confusion matrix corresponding to the comparison of the two preprocessing options with (without) baseline cancellation.

| $\Delta M_2$ | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 |
|--------------|---------|---------|---------|---------|---------|
| Class 1     | -139    | 231     | -108    | 30      | -14     |
| Class 2     | -38     | 0       | 41      | 0       | 0       |
| Class 3     | 0       | 0       | 55      | -63     | -14     |
| Class 4     | 0       | 0       | 0       | -7      | 0       |
| Class 5     | -19     | 0       | -11     | 17      | 18      |
7) FINAL CONFIGURATION

After the set of experiments, the only difference with the initial parameter values is the distance metric in the KNN classifier, which changes from Euclidean to Manhattan. The overall accuracy changes from 0.85 to 0.87. However, this is going to be dependent on the initial choices. For instance, our initial configuration included data normalization, and the methodology showed that without data normalization, the accuracy drops by about 7%. We also computed sensitivity, precision, and specificity for every class, obtaining subtle differences, with improvements, except for the precision of the ventricular (V) class, which improved from 39% to 56%, meaning that fewer beats from other families are misclassified as ventricular. Even if the improvements for this particular case are modest, the methodology proves to be useful for parameter tuning in terms of the overall accuracy of the system and prevents overfitting due to the inherent imbalance of the problem.

We performed an analysis with LIME, as we did with the surrogate database, to interpret the results of our model with a real multiclass dataset. Figure 16 illustrates an example of two beats from the multiclass database, where Figure 16 (a) shows a beat labeled as Class 1 and correctly classified as Class 1, and Figure 16 (B) shows a beat labeled as class 5 but erroneously classified as Class 4. For each example, the top panel shows the beat, the medium panel shows the signal samples that LIME determines are being used to classify the example, and the bottom panel shows the same samples weighted by their importance in the classification. We have reviewed several examples of identifying that consistently the same beat samples are representative of each class in many cases.

We also obtained the t-SNE representation with the multiclass dataset. Figure 17 (a) shows how the different classes of multiclass data are distributed in a low dimensional space. There is a strong overlap between Class 2 and the other classes, making difficult the visualization of the remaining classes and also the interpretation of this representation. Figure 17 (b) shows the same visualization for the biclass dataset. Both classes overlap, making it difficult to differentiate them and showing that this is an intrinsically hard classification problem.

D. BICLASS DATASET

We conducted similar experiments for the biclass regrouping. After comparing the different preprocessing options, the selected combination of parameter values is described next.

$$\theta_{FIL}^2 = 40 \text{ Hz as the second preprocessing option, denotes } $$
$$\theta_{FIL}^2 = 50 \text{ Hz is a better preprocessing option. Also, } $$
$$\text{comparing the values of the differential confusion matrix yielded better results when choosing } \theta_{FIL}^2 = 60 \text{ Hz as the cut-off frequency, specifically improving the first class over the other classes. After comparing different options, we selected } \theta_{FIL}^2 = 60 \text{ Hz as the cut-off frequency.}$$

### Table 9.
Confusion matrices for the initial configuration (a) and final configuration (b) of preprocessing options. (c) Metrics for the initial and final configuration.

|       | Class 1 | Class 2 |
|-------|---------|---------|
| Class 1 | 34934  | 2281 |
| Class 2 | 5357   | 5642  |

(a)

|       | Class 1 | Class 2 |
|-------|---------|---------|
| Class 1 | 36701  | 514 |
| Class 2 | 4799   | 6200  |

(b)

|       | Accuracy | Precision | Sensitivity | Specificity |
|-------|----------|-----------|-------------|-------------|
| Initial | 0.84     | 0.71      | 0.51        | 0.94        |
| Final   | 0.89     | 0.92      | 0.56        | 0.99        |

(c)

First, not normalizing was better in this case. As seen in Figure 15 (a), normalizing (not normalizing) the data is the first (second) option. Second, we selected to use all the beats from each class per patient, as seen in Figure 15 (b), as far as this option is better than the first option implemented with $N_m = 50$. Third, the distance used in the KNN classification algorithm was $d_m$, as seen in Figure 15 (c), compared with $d_c$. Also, the number of neighbors to be used was selected to be $k = 2$, as seen in Figure 15 (d). Finally, baseline cancellation of the ECG signal is shown to be preferable in Figure 15 (d), and $\theta_{FIL}^2 = 70 \text{ Hz for low-pass filtering in Figure 15 (f).}$

Table 9 shows the confusion matrices for the initial (a) and final (b) preprocessing options. If we define Class 1 beats as negative cases and Class 2 beats as positive cases, Table 9 (c) shows the accuracy, precision, sensitivity, and specificity metrics for the initial and final preprocessing configurations. It should be noted that the accuracy provided by the classifier is again modest, as far as we chose to use a simple classifier in a very basic configuration. However, these overall results show several positive results about the proposed DBA index and the differential confusion matrices used. On the one hand, the beat detection system is able, to some extent, to retrieve some relevant information and to give some class separability. This represents a noisy environment for any performance test, as the class overlapping is strongly present. However, the proposed method is able to guide the signal processing and ML tuning to a statistically principled increase in performance even in these non-favorable conditions. On the other hand, the increase in overall accuracy (DBA index) can be checked to correspond to a specific improvement in the confusion matrix elements (i.e., increase in the diagonal elements and decrease in the non-diagonal elements), even in severe conditions of class unbalance, which is often present in beat classifiers for ECG systems when using ML algorithms. Our method allows us to deal with this kind of scenario and to set an appropriate choice of settings for the signal preprocessing and free parameters for ML algorithms.
VI. DISCUSSION

The main objective of this work is to suggest a methodology to evaluate the impact of the signal preprocessing and ML options on the final beat families classification results by using a statistically principled DBA index and the differential confusion matrices. Notice that this methodology conveys extra computational load during the design of the classification system but not afterward when the system is in use. We analyzed the beat family identification problem using ML algorithms and the ECG preprocessing impact on the final classification error using a simple classifier, namely, the KNN algorithm. Generalization in ML systems requires a large enough number of examples in each beat group and patient diversity to be included as the data sources. It can be sensitive to the intrinsic strong imbalance in the different classes. We selected this ML algorithm because it has been extensively used in the literature, and its performance is enough to analyze its generalization capabilities on different conditions. This algorithm has just one hyperparameter to tune, so it is appropriate to focus on evaluating the previous
FIGURE 16. LIME examples for multiclass dataset. (a) The beat is labeled as Class 1 and it is classified as Class 1, (b) The beat is labeled as Class 5, but it is classified as Class 4. For each example, the top panel shows the beat, the medium panel shows the signal samples that LIME determines are being used to classify the example, and the bottom panel shows the same samples weighed by their importance in the classification.

FIGURE 17. (a) t-SNE representation of multiclass database; (b) t-SNE representation of biclass database.

stages. The ML requirements on databases for their successful generalization could be different from the ones offered by Holter benchmarking databases, which focus on performance examples in different signal conditions, like segments with specific types of noise, pacemakers, or some non-sinus beats. This represented an opportunity to scrutinize the usefulness of the method to establish clearer comparisons among the performance when using different ML methods in the family beat classification stage.

We have validated and compared the proposed methodology using a custom-surrogate dataset generated with simple definitions. Given its use in the ML literature, we also used the MIT-BIH database in our experiments. We regrouped the family labels into two sets, multiclass and biclass. To achieve the proposed objectives, we studied preprocessing in different stages, noting that the preprocessing options affect the classification of the beats morphology for the two regrouping data differently. In other words, comparing two different preprocessing options is not straightforward. One particular setting may improve the classification for some family beats classes and worsen the classification for other family beats classes. Here special attention must be paid to the intrinsic strong imbalance between the different classes. If we only consider the classification error, the algorithms will tend to learn the details of the majority-beat family classes at the expense of a poor classification of the minority-beat family classes. The
use of two classes in the experiments had a twofold motivation. On the one hand, from a benchmarking viewpoint, we wanted to test how our proposal for analyzing the quality of the beat classifiers worked in a non-hardly unbalance scenario, as far as results on multiclass labeling dataset were shown to be strongly biased in terms of apparently improved accuracy just by increasing the detection towards the most populated class. On the other hand, according to our clinical coauthors, it is true that, ideally, it would be desirable to have a beat classifier available to distinguish among multiple classes. Nevertheless, in clinical practice, it is accepted that just a separation between two classes would be useful, as far as it can be reliable. In many existing works, publicly available databases, like the MIT-BIH Arrhythmia Database (often) or the PTB Diagnostic ECG Database (sometimes). However, they are undoubtedly valuable but should still be noted as not being initially designed for training ML systems, as suggested before [49]. The small number of records of some patients and the unbalanced classes are examples of this. The system generalization must be taken into account, as beats in the same patients are highly repetitive, so complete patients should be left separate to perform the validation, and the test stages, which in too many cases is not reported, and very likely is not done. This could be the cause for the extremely high-performance values obtained from systems in the ML literature for ECG processing, sometimes due to the same bias in the machine training. Another problem that should be pointed out is that some of these works are not addressed with the support of cardiologists in the research team, which leads to loose definitions of the clinical applications. For the multiclass regrouping, the final result can be summarized as follows. Data normalization improved overall results by 7%. When balancing, we chose $N_m = 50$ as the maximum number of beats selected from each class per patient because minority classes improved by about 3% their accuracy. Baseline cancellation was found appropriate according to the confusion matrix details. In the KNN algorithm, we selected $k = 2$ and Manhattan distance. Finally, the chosen cut-off frequency in the low-pass filter was 60 Hz because it improved the performance for all classes concerning the other scrutinized cut-off frequencies. The final overall accuracy is 0.87.

For the biclass regrouping, the final results can be summarized as follows. Without normalizing, the data accuracy improves by about 6%. Beat number balancing did not improve the performance. In the KNN algorithm, we selected $k = 2$ and Manhattan distance. Baseline cancellation was found appropriate, and the best cut-off frequency in the low-pass filter was 70 Hz. With the proposed procedure, all metrics have improved. For instance, the precision increased from 0.71 in the initial configuration to 0.92 in the final design, meaning that a smaller number of supraventricular beats are misclassified as ventricular beats. The absolute overall accuracy is 0.89.

Several implications can be drawn from our study to be accounted for in future fieldwork. First, from the precedent results, it can be seen that preprocessing stages can significantly affect the overall performance. This aspect is often hindered by works on ML methods for beat families classification. Future jobs in this application should consider the ML and signal preprocessing aspects. Second is the impact of the imbalance in the minority classes and the out-of-sample strategy that is (or is not) used in preceding works. In [49], a review of recent works using deep learning algorithms for this problem showed that most of the current jobs in the literature report overfitting results. The out-of-sample strategy should be carefully designed not to split data beats from the same patient in training and the test set, as far as the independence of the datasets is lost. Apparently high accuracy is obtained by many methods. The muscular imbalance in some of the classes makes still more sensitive the learning process out-of-sample design limitations. Whereas databases such as the MIT-BIH have been an excellent resource for Holter system benchmarking, they should be handled with methodological caution when training and testing ML algorithms.

Several considerations should be taken into account regarding the scope of the present paper. On the one hand, several works have been delivered over the last few years, pointing to the risk of the subtle and unnoticed presence of overfitting on the part of the literature on ECG classification, especially including works with the MIT-BIH database. In a precedent work by our group [49], a detailed analysis was made in Section 6, Applications in ECG Processing, and especially in Subsection 6-3-Open Issues for Deep Learning in ECG. As pointed out, some works made a train-validation-test partition which did not ensure that beats from the same patient were constrained to be in only one of these sets, and its effect sometimes increased the apparent performance. In addition to our group, other authors have pointed out this risk in delivering results in the context of ECG analysis for atrial fibrillation detection in recent years [68], [69]. These references and others motivate the need for cut-off tests to guide the design of signal processing and machine learning stages on beat classifiers. On the other hand, what we wanted to propose is a non-parametric cut-off test for this purpose and to show that it can be used with an advantage to improve the design conditions for a beat classifier. In this setting, we decided not to include several classifiers in work to make it easier to follow and to avoid the impression of being a work on classifier benchmarking. Instead, the use of KNN was motivated by its simplicity. Even though its performance is so moderate, the method allowed us to improve the beat classifier by tuning with the proposed non-parametric cut-off tests.

Many interesting future research lines are open, given the results and conclusions from this work. For instance, to validate the proposed methodology with other accurate databases, apply different novel and good algorithms, such as DL and dictionary learning, and use feature interpretability techniques on ML classifiers with improved performance.
VII. CONCLUSION
Whereas ML offers the field of beat recognition systems an excellent set of tools, several cautions should be taken to avoid risks. Overall, cooperation with cardiologists should be strongly recommended, as well as consideration of out-of-sample training strategies. Unbalance in classes is intrinsic to the beat classification problems. Either out-of-sample design or algorithms exhibiting robustness to imbalance should be pursued. In this setting, data augmentation approaches based on replicating beats can be especially disadvantageous. Excellent databases created for Holter and cardiac monitor benchmarking, such as the MIT-BIH arrhythmia database, should be used with caution when creating ML-based family beat classifiers. New public databases should consider benchmarking, such as the MIT-BIH arrhythmia database, an excellent set of tools, several cautions should be taken to avoid risks. Overall, cooperation with cardiologists should be especially advantageous. Whereas ML offers the field of beat recognition systems an excellent set of tools, several cautions should be taken to avoid risks. Overall, cooperation with cardiologists should be especially advantageous.
D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Appl. Comput. Harmon. Anal.* vol. 26, no. 3, pp. 301–321, May 2008.

M. Butkuviene, A. Petrenas, A. Solosenko, A. Martin-Yebra, V. Marozas, “Considerations on performance evaluation of atrial fibrillation using combined linear and nonlinear features of ECG signals,” *Comput. Methods Programs Biomed.*, vol. 100, pp. 52–63, Apr. 2016.

L. Sormo, A. Petrenas, M. Henrikkson, and V. Marozas, “Letter regarding the article ‘Detecting atrial fibrillation by deep convolutional neural networks’ by xia et al.,” *Comput. Biol. Med.*, vol. 100, pp. 41–42, Sep. 2018.

M. Butkuviene, A. Petrenas, A. Solosenko, A. Martin-Yebra, V. Marozas, and L. Sormo, “Considerations on performance evaluation of atrial fibrillation detectors,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 11, pp. 3250–3260, Nov. 2021.
RAÚL P. CAULIER-CISTERNA received the dual Ph.D. degree in multimedia and communications from Universidad Rey Juan Carlos and Universidad Carlos III of Madrid, Spain, in 2018, with specialization in bio-engineering and biomedical signal processing. From 2009 to 2012, he was with the Department of Signal Theory and Communications, Universidad Carlos III of Madrid. From 2013 to 2018, he was with the Department of Signal Theory and Communications and Telematics and Computational System, Universidad Rey Juan Carlos of Madrid. He is currently working as a Postdoctoral Fellow with the Biomedical Imagine Center, Pontificia Universidad Católica de Chile. His research interests include machine learning methods for signal and image processing, arrhythmia mechanisms, robust signal processing methods for cardiac repolarization, intelligence artificial algorithms, and non-invasive near infrared spectroscopy in spinal cord.

ARCADI GARCÍA-ALBEROLA received the M.D. and Ph.D. degrees from the Universitat de Valencia, Burjassot, Spain, in 1982 and 1991, respectively. Since 1993, he has been a Cardiologist and a Professor in medicine at Hospital Universitario Virgen de la Arrixaca and Universidad de Murcia, Murcia, Spain, where he is currently the Director of the Arrhythmia Unit of Cardiac Electrophysiology. His research interests include repolarization analysis, arrhythmia mechanisms, and cardiac signal processing.

JOSÉ LUIS ROJO-ÁLVAREZ (Senior Member, IEEE) received the B.Sc. degree in telecommunications engineering from the Universidade de Vigo, in 1996, and the Ph.D. degree in telecommunications engineering from the Universidad Politécnica de Madrid, in 2000. He is currently a Professor at the Department of Signal Theory and Communications, Universidad Rey Juan Carlos, Spain. He has coauthored more than 140 international papers and has contributed to more than 180 conference proceedings. His research interests include statistical learning methods for signal and image processing, arrhythmia mechanisms, robust signal processing methods, and data science for arrhythmias.

* * *