Machine Learning for Real-Time Heart Disease Prediction

Dimitris Bertsimas, Luca Mingardi, and Bartolomeo Stellato, *Member, IEEE*

***Abstract*— Heart-related anomalies are among the most common causes of death worldwide. Patients are often asymptomatic until a fatal event happens, and even when they are under observation, trained personnel is needed in order to identify a heart anomaly. In the last decades, there has been increasing evidence of how Machine Learning can be leveraged to detect such anomalies, thanks to the availability of Electrocardiograms (ECG) in digital format. New developments in technology have allowed to exploit such data to build models able to analyze the patterns in the occurrence of heart beats, and spot anomalies from them. In this work, we propose a novel methodology to extract ECG-related features and predict the type of ECG recorded in real time (less than 30 milliseconds). Our models leverage a collection of almost 40 thousand ECGs labeled by expert cardiologists across different hospitals and countries, and are able to detect 7 types of signals: Normal, AF, Tachycardia, Bradycardia, Arrhythmia, Other or Noisy. We exploit the XGBoost algorithm, a leading machine learning method, to train models achieving out of sample F1 Scores in the range 0.93 – 0.99. To our knowledge, this is the first work reporting high performance across hospitals, countries and recording standards.**

***Index Terms*— Boosting, ECG, Machine Learning, Arrhythmia**

I. Introduction

Despite the continuous development of medical practices, heart-related diseases are still the leading cause of death in the United States [13]. Atrial Fibrillation (AF) is among the most common ones, as it affects 1-2% of the general population, causing hundreds of thousands of deaths every year, as it can lead to a stroke, heart failure or coronary artery disease [14]. Machine Learning (ML) techniques are becoming more and more accepted in the world of healthcare as a support to traditional ways of disease detection. In fact, algorithms can be leveraged to process a sizeable amount of data in a fast and accurate way, allowing to get non-obvious insights directly from the observations.

One of the problems of AF detection is that it is often asymptomatic (it is incidentally identified in 30–45% of patients who had an electrocardiogram for unrelated reasons [19]) and trained personnel is required to spot the disease from electrocardiograms (ECG). Unfortunately, if AF is not promptly recognized and treated, it can lead to a fatal event, such as a stroke. Similarly, Tachycardia (excessively fast heart rate) and Bradycardia (excessively slow heart rate) are common heart diseases. Despite being less dangerous than AF, they can lead to serious complications, such as heart failure, if left untreated.

Given the severity and occurrence of heart diseases, procedures to analyze ECGs have been in place for a long time. A breakthrough algorithm for QRS complex detection was published in 1985 [27], starting the era of machine learning analysis to detect arrhytmia from ECGs, also thanks to the MIT-BIH [24] made available since 1980 by Physionet. The availability of such data has inspired many works in the literature [30], [26], but some limitations of these works come from the unrealistic nature of their datasets: there are only Normal or AF ECGs (there can be other types of signal); only clean signals are considered; the sample size is small. More recent projects leverage Deep Neural Networks to extract non-linear features from the ECGs [29], [6], [16]. In the first project, sponsored by Apple for the AF detector in its Watch products, the authors work on a dataset of 400 thousand patients and achieve Positive Predictive Value of 0.84, but they don't disclose the details of the model used. In the second project, the authors create an AF detector by analyzing 1 million signals with 12 leads and 10 seconds long, achieving an $F1$ Score of 0.45. In the third one, the authors build a Convolutional Neural Network to detect 12 heart abnormalities from 91 thousands 12-leads signals of various length, achieving AUC of 0.97 and $F1$ Score of 0.84. The data for these projects is not available to the public, thus complicating an objective comparison. In [33], the authors propose their analysis on a proprietary dataset, containing more than 40 thousand patients, and leverage Gradient Boosting Trees to achieve an $F1$ Score of 0.97. Finally, in [8] the authors develop a neural network framework to analyze a proprietary dataset containing 55 types of arrhythmias and more than 32 thousand patients, achieving 0.86 $F1$ Score. The training datasets from the last two projects are available online.

In this work, we present a novel procedure to accurately detect heart diseases in real-time from the analysis of short single-lead ECGs (9-61 seconds). We observe the characteristics of ventricular response and analyze the predictability of the inter-beat timing of the QRS complexes [5] (see Figure 1) in the ECG to detect irregular patterns in the data. Specifically, we extract four groups of features that we use for the predictions: time and non-linear domain, distance-based and time series characteristics. The model is meant to be used as a fast detector for heart diseases, able to recognize various types of outcomes: Normal, AF, Tachycardia, Bradycardia, Arrhythmia, Other (label to indicate other types of heart anomaly) and Noisy (can't be classified, and needs to be recorded again). If an anomaly is detected, the patient should be visited by a specialist to assess the stage of the disease and possible treatments.

We leverage the XGBoost algorithm [10], a leading machine learning method, to train and evaluate our models on three different datasets, achieving strong out of sample performance ($F1$ Score $\geq$ 0.94). Then, we test the performance of our models when used as predictors accross datasets and we achieve similar results ($F1$ Score $\geq$ 0.93).

II. Data

We focus our work on three main datasets. The first one comes from the 2017 Challenge from Physionet [12], which consists of a collection of 8828 recordings together with the corresponding labels (Normal, AF, Other and Noisy) given from expert cardiologists. This dataset has been recorded through AliveCor [3], a portable device able to record electrocardiograms. The second one comes from the 2019 Tianchi Hefei High-Tech Cup ECG Human-Machine Intelligence Competition [2], and contains 20019 observations across 5 different labels (Normal, Tachycardia, Bradycardia, Arrhythmia, AF) annotated by expert cardiologists. The third one comes from Chapman

Manuscript submitted on October 10th 2020

D.B. is with the Sloan School of Management, Massachusetts Institute of Technology (e-mail: dbertsim@mit.edu).

L.M. is with the Operation Research Center, Massachusetts Institute of Technology (e-mail: lucam@mit.edu).

B.S. is with the Department of Operation Research and Financial Engineering, Princeton (e-mail: bstellato@princeton.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3066347, IEEE Journal of Biomedical and Health Informatics

University and Shaoxing People's Hospital [1] and contains 10646 observation with the corresponding labels (Bradycardia, Normal, AF, Tachycardia) given by expert cardiologists.

There are significant differences among the datasets. The first one is a collection of recordings of people from the United States, while the other two are recorded among the Chinese population. Moreover, the first dataset is recorded through a wearable device at 300 Hz (samples per second) and have a length ranging between 9 and 61 seconds. The ECG signals of the other two are 10 seconds long and are recorded in hospital with professional machines at 500 Hz. For this reason, the first one comes with a single lead recording, while the other two have the usual 12 ECG leads. However, we believe that our model can be most useful when used to detect heart problems in real time, through the use of portable devices. Thus, in order to simulate a real time recording, we kept only lead II (the one containing the best QRS recordings) in the signals from the two Chinese datasets, and proceed with a unified pipeline for our analysis. These two datasets also provide demographics information (such as Age, Gender) of the patients, which is very valuable because heart diseases are often correlated with such characteristics.

III. Pipeline

A. Signal Processing

While an ECG is recorded, there are a number of different factors that can impact the quality of the signal, such as the movement of the patient or the powerline noise coming from the electric component of the machinery. Thus, preprocessing the original recording is a necessary step to eliminate the noise in the data. We apply two filters: butterworth highpass [7] (lowcut = 0.5 Hz) and band-pass [9] (cutoff = 0.05 Hz), and then we scale the signals to have zero mean and unit variance. These operations are performed through the Scikit-Learn [28] and SciPy [32] implementations in Python.

B. Feature Extraction

Our approach is based on the extraction of features (see Appendix for full list of features) directly from the recorded ECGs. Specifically, we develop a novel method to extract in real time 110 features, which can be divided in four main groups, as described in Table I:

TABLE I
Summary of the Extracted Features.

Group	Description	Count
1	Time-domain indices of HRV	12
2	Nonlinear-domain indices of HRV	7
3	Distance-based features	40
4	Time series ECG characteristics	51

First, we find the R peaks in the QRS complexes of each ECG (Figure 1) by implementing the Pan-Tompkins algorithm [23]. For the first group of features, we analyze the QRS complex of each signal and we calculate statistics (e.g. mean, standard deviation, proportion of abnormal intervals) related to the interval between subsequent R peaks. For the second group, we calculate non-linear features related to the Heart Rate Variability (HRV) of the patients (e.g. Cardiac Sympathetic Index [31]). For the third group, we detect the Q, S, P and T waves by inspecting the near surroundings of the R peaks. In fact, once the R peaks are accurately identified, the other four points can be easily found, as Figure 1 intuitively suggests. Point Q and S are downward deflections of the QRS complex respectively immediately before and after the R peak. On the other hand, point P and T are smaller peaks occurring respectively right before point Q and right after point S. The area of inspection around the R peaks has been determined through extensive experiments to find the most accurate results. Specifically, let A, B and C be consecutive R peaks, α be the distance between A and B, and β be that between B and C: point P corresponds to the maximum value in segment $B - 0.35\alpha$, point Q corresponds to the minimum value in segment $B - 0.1\alpha$, point S corresponds to the minimum value in segment $B + 0.1\beta$ and point T corresponds to the maximum value in segment $B + 0.35\beta$. Other methods such as moving averages and derivative analysis have been explored throughout the work, proving to be less accurate, much slower and thus less viable. Once the 5 points of the PQRST complex are found, we calculate the pair-wise vertical and horizontal distances between each combination. Then, we calculate the average, median, minimum, maximum, standard deviation for each of them, for a total of 109 features. For the fourth group, we leverage the TSFRESH python package [11], and we extract 742 features related to the characteristics of an ECG as a time series. Despite increasing the computational complexity, the last group accounts for significant improvement in performance (up to 7%), thanks to its focus on time series oriented features. As a final result, we obtain a dataset composed of 880 features for each signal.

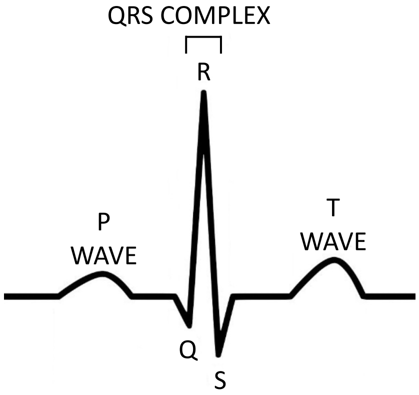

Fig. 1. Visualization of a QRS complex in an ECG.

C. Modeling Approach

The problem under consideration is multi-class classification, so that it is not possible to directly leverage the usual methods for binary classification. However, the XGBoost algorithm [10] allows to set as its objective the softmax function:

$$q_i(z) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \tag{1}$$

which outputs the probability that a given observation belongs to each of the labels z_i in the dataset in a vector format. Then, the classifier minimizes the cross-entropy loss between the real distribution of labels p and the estimated probabilities q:

$$\mathcal{L}(p, q) = -\sum_x p(x) \log q(x), \tag{2}$$

where x is the set of all the observations in the data, and p is the one-hot encoding of the true label associated to each observation.

We train 5 different models: one for each available dataset (3), and then we exploit overlapping labels among datasets to train 2 models and assess the predictive performance across data sources. Specifically, we have overlapping labels between Tianchi and Chapman data regarding Normal, Tachycardia and Bradycardia, and between Physionet and Chapman data regarding Normal and AF (Tianchi data has too few AF observations). In the case of Tianchi and Chapman, we leverage the features available in the data, in addition to the 110 we extract with our procedure (e.g. Age, Gender, T-Offset, P-Offset etc.). Due to the different set of labels present across datasets, the two cross-dataset models are trained on the subset of overlapping labels in the training data and evaluated only on the observations having these labels in the testing data. Table II describes in detail the 5 models, listing the dataset specific features that are used in addition to the 110 we extract, and summarizing the patients, labels, training, testing, signal and sampling characteristics of each dataset.

D. Feature Selection

Inevitably, the features that we extract in Section III-B are highly collinear, thus it is extremely important to select a subset of them in order to improve the performance of each classifier and speed up the extraction process. We leverage the built-in feature importance method of the XGBoost algorithm, which ranks the features that have the highest explanatory power with respect to the outcome. Specifically, for each of the 5 tasks presented in Section III-C, we train a XGBoost model with default parameters. Then, for each of these models, we select the top 50 features identified by the algorithm. Finally, among these 250 features, 140 are duplicates, thus leaving with 110 features for the final model. Finally, we notice that the performance of the models doesn't improve by adding other features, finding that 110 is the minimum number of features not to have any drop in performance, accounting for a significant reduction in dimensionality from the initial set of 880 features extracted. After the final set of features is decided, we find the optimal set of parameters for each models according to the methods explained in Section III-E.

E. Methods

In our work, we leverage the XGBoost algorithm [10] to train our models and the Optuna optimization framework [4] to tune its parameters. Finally, we use SHAP [20], [21], [22] to explain which features have the most explanatory power for each label present in the training data.

a) XGBoost: XGBoost [10] is one of the most popular and performing tree-ensemble methods for binary classification. It is based on an iterative procedure that leverages a large number of trees. Its strength lies in the iterative correction procedure on which it is based, so that new trees are added to correct the errors made by earlier trees, allowing the model to handle better the harder cases. There are many hyperparameters determining its performance, the most important of which are: depth of trees, number of trees and learning rate. The tuning of the parameters of the algorithm have a significant impact on its performance, and a proper procedure can be followed to avoid the overfitting on the training data: there is usually a trade-off between the performance on the training set and that on an external testing set. Proper tuning allows to achieve a strong performance on both datasets, if there is enough explanatory power in the variables of the dataset.

In this work, we tune seven parameters. The maximum depth of a tree determines the maximum number of nodes that can exist between the root node and the farthest leaf in the tree: it takes positive integer values, with large ones usually leading to overfitting. The number of estimators controls the number of trees to fit in the training: it takes positive integer values, with large ones usually lead to overfitting. The learning rate η controls the weighting factor for corrections by new trees: it takes values between 0 and 1, with values closer to 0 determining fewer corrections for each tree. The parameter γ determines the minimum loss reduction required to make a further partition on a leaf node of a tree: it takes positive values, with larger ones defining a more conservative model. The parameter λ is the $L2$ regularization on the weights of the features: it takes positive values, with the larger ones shrinking the weights, thus making the model more conservative. The parameter α is the $L1$ regularization on the weights of the features: it takes positive values, with the larger ones driving to 0 the weights, defining a more conservative model. Minimum child weight is the minimum Hessian weight required to create a new node: it takes positive values, with higher ones making the model more conservative. All remaining parameters are set to their default values.

b) Optuna: Optuna [4] is a leading optimization framework leveraging Tree-structured Parzen Estimator (TPE) to optimize an objective function over a defined parameter space. Each of the 5 models is trained independently following three steps. We choose the range (the same for all the models) of possible values for the seven parameters explained above, we define the objective function to maximize as the average 70-folds cross validation $F1$ Score , and finally we leverage multiple cores to maximize the objective function over 500 iterations.

c) SHAP: SHAP [20], [21], [22] is a method to interpret Machine Learning models through a game theory approach. This method is helpful to dig further into how the final predicted, so that it highlights the most important features and explains how they drive the results in an understandable way.

The analysis has been performed using Python 3.7.5.

F. Performance Evaluation

In order to assess the performance of our procedure, for the Physionet data we exploit the external 300 ECGs provided as validation set, while for the other models we divide the data 70% training and 30% testing set. Then, we report the Accuracy, Precision, Recall and F1 Score calculated on each test dataset. For example, in the case of Atrial Fibrillation, we calculate the metrics as:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Number of Predictions}} \tag{3}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \tag{4}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{5}$$

$$F_{1a} = \frac{2Aa}{A+a} = \frac{2(\text{precision} \cdot \text{recall})}{\text{precision} + \text{recall}}, \tag{6}$$

where Aa is the number of correct AF predictions, A is the number of predicted AF and a is the number of true AF. The same calculation is performed for the all the other classes, and we report both the arithmetic and weighted mean of the $F1$ Scores, for each model that we train. The weighted $F1$ Score is calculated as:

$$\text{Weighted } F_1 \text{ Score} = \frac{\sum_i F_{1i} \cdot N_i}{\sum_i N_i} \tag{7}$$

with i = 1, ..., n, with n being the number of classes and N_i being the number of observations for class i. The accuracy is calculated as the number of correct predictions over the total number of predictions. In all the three datasets there is only one recording per patient, thus it is not possible to have an unfair evaluation during the training-testing step.

G. Output Calibration

Confidence calibration is a particularly relevant problem in a healthcare setting like the one addressed in this manuscript: when a Machine Learning model makes a prediction, it is important that this output can be trusted. For example, if the model makes 100 label predictions with confidence of 0.9, 90 of them should be

TABLE II
MODEL AND DATA SUMMARY.

Model	Additional Features	Patients	Labels	Training	Testing	Signal Len.	Sampl. Freq.
Physionet	None	8828	Norm, AF, Other, Noisy	8528	300	9-61s	300Hz
Chapman	Age, Gender, T-Offset, P-Offset VentricularRate, AtrialRate, QRSDuration QTInterval, QTCorrected, RAxis, TAxis TOffset, QRSCount, QOnset, QOffset	10646	Norm, AF, Tachy, Brady	7452	3194	10s	500Hz
Tianchi	Age, Gender	20019	Norm, AF, Tachy, Brady, Arrhy	14013	6006	10s	500Hz
Chapman-Tianchi	None	28080	Norm, Tachy, Brady	8421	19659	10s	500Hz
Physionet-Chapman	None	10484	Norm, AF	6034	4450	9-61s	300-500Hz

correct. While perfect calibration is not usually achievable in a real setting, there are metrics to quantify how reliable a model is. In our work, we follow the procedure highlighted in [15] to calibrate each model's output using Temperature Scaling (the details of which can be found in the paper), inspect the relationship between accuracy and confidence, and calculate the corresponding Expected Calibration Error (ECE). To estimate the expected accuracy, the predictions are grouped in M interval bins (10 in this case) and we calculate the accuracy of each bin B_m as:

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i) \tag{8}$$

where $\hat{y}_i$ and y_i are the predicted and true class labels for sample i. The confidence of bin B_m is calculated as:

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{9}$$

where $\hat{p}_i$ is the confidence for sample i. A perfectly calibrated model would have $\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for all $m \in (1, ..., M)$. The Expected Calibration Error can be approximated by taking a weighted average of the difference between the accuracy and the average of each bin [25]:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|, \tag{10}$$

where n is the number of samples. The difference between acc and conf is called calibration gap and is represented by the red bars of a reliability plot (Figure 7, 8, 9, 10, 11 in the Appendix).

Table III reports the improvement in Expected Calibration Error when the output is calibrated using Temperature Scaling (we implement this method in python leveraging [18]). This procedure improves significantly the ECE for all the models but the Physionet one. This is due to the fact that the test set of this model is the smallest (only 300 observations). Overall, the five models prove to be very well calibrated on average and thus their predictions should be considered reliable.

TABLE III
DELTA ECE CALIBRATION.

Paper	Uncalibrated	Calibrated
Physionet [12]	0.030	0.035
Chapman [1]	0.026	0.006
Tianchi [2]	0.007	0.001
Chap-Tian	0.004	0.0008
Phys-Chap	0.024	0.020

IV. RESULTS

In this section we propose the out of sample performances of the 5 models that we have trained following the pipeline explained above, and the corresponding 10 most important features. From Table IV it is possible to observe the strong perfomance of the model trained on Physionet data (Weighted Average $F1$ Score 0.94). Table V is useful to inspect where the model is making errors. In this case, the models makes the most errors in the Other and Noisy class. The confusion associated to the Other class comes from the definition of it, as it includes different (non-specified) heart anomalies, so that it is tough for the model to learn the exact characteristics of it. On the other hand, when the model is wrong and predicts Noisy, it doesn't lead to a harmful decision, given that after a Noisy prediction, the model would ask to record another signal, which would then be classified another time. Figure 2 displays the most important features for this model. The CSI index is feature that explains the most when a signal is labeled as Normal or Other, and in fact is a popular index exploited in the literature [31] to identify anomalies in an ECG. The proportion of R-R intervals which are longer than 50 seconds is the feature that has the highest explanatory power for the AF label, which is in line with the fact that a person having Atrial Fibrillation has an irregular heart beat. Finally, The Spectral Centroid feature is able to capture quick variations in the recording, making it an ideal feature to spot a Noisy signal. In a real setting, this model can be particularly useful because of the presence of the Other class, which is meant to represent more generally the characteristics of a ECG signal of a problematic heart condition. In fact, in the occurrence of an undefined or new heart condition, this model would be able to detect it, so that the patient can be directed to a specialized professional to further investigate and cure the problem.

TABLE IV
METRICS TABLE PHYSIONET.

Description	Precision	Recall	F1 Score	Count
Normal	0.97	0.97	0.97	150
AF	0.98	0.92	0.95	50
Other	0.97	0.89	0.93	70
Noisy	0.75	1.00	0.86	30
Accuracy			0.94	300
Arith. Avg	0.92	0.94	0.93	300
Weight. Avg	0.95	0.94	0.94	300

TABLE V
CONFUSION MATRIX FOR THE MODEL TRAINED ON PHYSIONET DATA.

Physionet	Pred. Normal	Pred. AF	Pred. Other	Pred. Noisy
True Normal	145	0	0	5
True AF	0	46	2	2
True Other	4	1	62	3
True Noisy	0	0	0	30

Table VI shows that the model trained on Chapman data is extremely

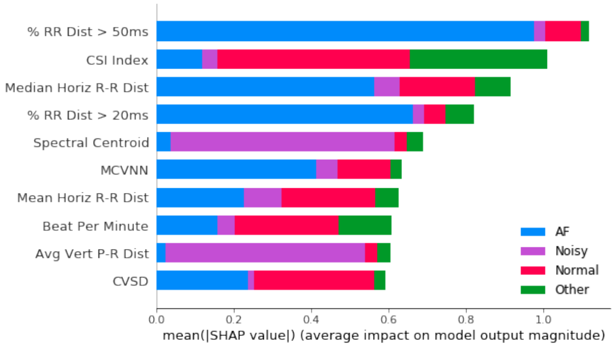

Fig. 2. Feature Importance Physionet.

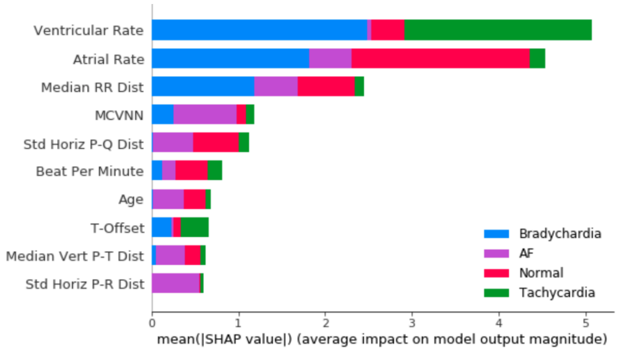

Fig. 3. Feature Importance Chapman.

accurate over the four classes under observation (Weighted Average $F1$ Score 0.96). Table VII shows where the model tends to predict incorrectly. In this case, the model is pretty consistent over the four classes, with most of the errors being a mislabeling between AF and Tachycardia. However, anytime the model predicts anything different from Normal, we suggest that the patient is visited from a trained doctor who can assist the patient. Figure 3 displays the most important features for this model. Atrial Rate is the most important feature to identify a Normal signal, and indeed this is an important factor to consider when analyzing an ECG. In the same fashion, The Ventricular Rate, is the key feature to identify Bradycardia and Tachycardia. These two features are available directly in the Chapman data, but are highly correlated with some of the features extracted with our pipeline (e.g Beat per Minute). Finally, MCVNN is the best feature to identify Atrial Fibrillation. It is calculated as the median absolute deviation of the horizontal R-R distances divided by the median of the absolute differences of successive horizontal R-R distances. The predictive power of these features come from their ability to capture anomalies in successive heart beats.

TABLE VI
METRICS TABLE CHAPMAN.

Description	Precision	Recall	F1 Score	Count
Bradychardia	0.98	0.99	0.99	1167
Normal	0.96	0.97	0.97	667
AF	0.93	0.91	0.92	668
Tachycardia	0.95	0.95	0.95	692
Accuracy			0.96	3194
Arith. Avg	0.96	0.95	0.96	3194
Weight. Avg	0.96	0.96	0.96	3194

TABLE VII
CONFUSION MATRIX FOR THE MODEL TRAINED ON CHAPMAN DATA.

Chapman	Pred. Brady	Pred. Normal	Pred. AF	Pred. Tachy
True Brady	1157	2	8	0
True Normal	4	646	14	3
True AF	12	16	607	33
True Tachy	3	6	25	658

Table VIII shows that the model trained on Tianchi has almost very accurate performance (Weighted Average $F1$ Score 0.99). Table IX shows that the model indeed makes almost no mistakes in its predictions. The errors associated to the Arrhythmia and AF classes come from the scarcity of such labels in the data. Moreover, given that AF is a form of Arrhythmia, it is not clear the difference between the two in the data, so that it is reasonable that the model is missing some of them. Figure 4 displays the most important features for this model. Beat per Minute is the feature that has the highest predicting power for the Normal and Tachycardia classes, which is related to the fact that Tachycardia is associated to an increased heart rate. Similarly, the median horizontal distance between successive R-R peaks is the best feature to identify Bradycardia, which is associate to a slow heart rate. Arrhythmia mostly relies on TINN, which is an approximation of the distribution of successive R-R intervals, for its identification. Finally, AF is identified through CVSD: the root mean square of the sum of successive differences in R-R intervals, divided by the mean of their lengths.

TABLE VIII
METRICS TABLE TIANCHI.

Description	Precision	Recall	F1 Score	Count
Normal	0.98	0.99	0.99	2850
Tachycardia	1.00	1.00	1.00	1469
Bradycardia	0.99	1.00	0.99	1579
Arrhythmia	0.83	0.51	0.63	76
AF	0.96	0.72	0.82	32
Accuracy			0.99	6006
Arith. Avg	0.95	0.84	0.89	6006
Weight. Avg	0.99	0.99	0.99	6006

TABLE IX
CONFUSION MATRIX FOR THE MODEL TRAINED ON TIANCHI DATA.

Tianchi	Pred. Norm	Pred. Tachy	Pred. Brady	Pred. Arr	Pred. AF
True Norm	2834	1	7	8	0
True Tachy	5	1464	0	0	0
True Brady	7	0	1572	0	0
True Arr	35	0	1	39	1
True AF	7	1	1	0	23

Table X shows that our model is able to achieve a very accurate performance also when evaluated on a dataset coming from a different source (Weighted Average $F1$ Score 0.99). In this case, we train on the Chapman data and evaluate on the Tianchi data. Table XI clearly shows that the model is reliably predicting across labels, with very few errors in the whole dataset. Figure 5 displays the most important features for this model. As one would expect, Beat per Minute and the median horizontal distance between successive R-R peaks are the most important features to identify the three labels under observation. In fact, the symptoms of Tachycardia and Bradycardia are increased and decreased heart rate, and these two features are a great proxy for it.

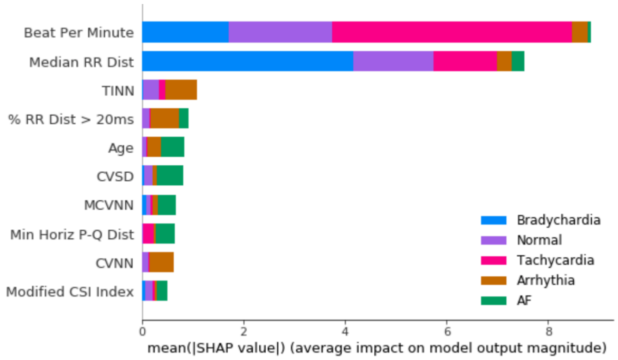

Fig. 4. Feature Importance Tianchi.

TABLE X
METRICS TABLE FOR THE CHAPMAN-TIANCHI MODEL.

Description	Precision	Recall	F1 Score	Count
Normal	1.00	0.99	0.99	9500
Tachycardia	0.99	1.00	1.00	4895
Bradycardia	0.98	1.00	0.99	5264
Accuracy			0.99	19659
Arith. Avg	0.99	0.99	0.99	19659
Weight. Avg	0.99	0.99	0.99	19659

TABLE XI
CONFUSION MATRIX FOR THE CHAPMAN-TIANCHI MODEL.

Chap-Tian	Pred. Normal	Pred. Tachy	Pred. Brady
True Normal	9375	26	99
True Tachy	11	4883	1
True Brady	20	1	5243

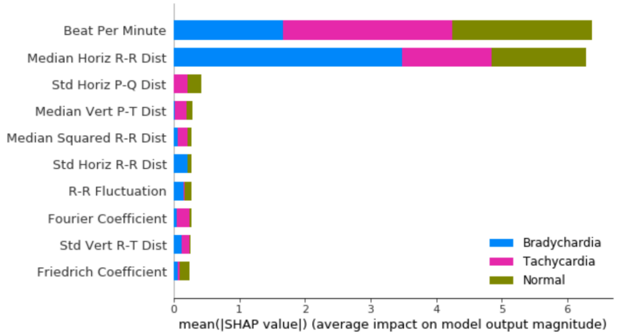

Fig. 5. Feature Importance model trained on Chapman and evaluated on Tianchi.

Finally, Table XII shows that the features that we extract are general enough to train a model that achieves high accuracy even when the training and testing datasets have very different characteristics (Weighted Average $F1$ Score 0.93). In fact, the Physionet data is recorded from a wearable device at 300 Hz and describes the characteristics of American people, while the Chapman data is recorded in a professional setting at 500 Hz (twelve leads are recorded but we only analyze lead II, as explained in Section II) and comes from the Chinese population. In this case, we train on the Physionet data and evaluate on the Chapman data. Table XIII shows balance in the errors between the two classes. Figure 6 displays the most important features for this model. Similarly to the model trained only on the Physionet data, the proportion of R-R intervals which are longer than 50 seconds is the feature that has the highest explanatory power for the AF label.

TABLE XII
METRICS TABLE FOR THE PHYSIONET-CHAPMAN MODEL.

Description	Precision	Recall	F1 Score	Count
Normal	0.94	0.92	0.93	2225
AF	0.92	0.94	0.93	2225
Accuracy			0.93	4450
Arith. Avg	0.93	0.93	0.93	4450
Weight. Avg	0.93	0.93	0.93	4450

TABLE XIII
CONFUSION MATRIX FOR THE PHYSIONET-CHAPMAN MODEL.

Phys-Chap	Pred. Normal	Pred. AF
True Normal	2036	189
True AF	130	2095

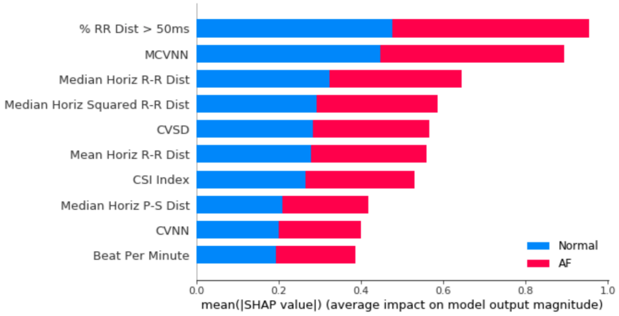

Fig. 6. Feature Importance model trained on Physionet and evaluated on Chapman.

The variance in performance across datasets (0.93-0.99) could be interpreted as confusing and potentially harming in a real setting. However, it is important to understand that such variance comes from the wide difference between the datasets. For example, in the case of the Chapman-Tianchi model (Weighted Average $F1$ Score 0.99), the labels are easier to predict, as it is known that Tachycardia and Bradycardia are easier to distinguish among themselves and with a Normal ECG. On the other hand, in the case of the Physionet model (Weighted Average $F1$ Score 0.94), the labels observed are much more challenging to predict and a drop in performance is to be expected. All the models trained in this work achieve a predictive performance similar or better than the ones that are presented in the corresponding original papers, thus confirming that it is not always possible to achieve a perfect result.

V. REAL-TIME ANALYSIS

The ultimate goal of our work is not to substitute the precious role of specialized professionals, but to provide an aid to them, accurately screening people with possible heart conditions which can be directed to such expert for deeper analysis. Thus, a key role in the viability and usefulness of our tool is its time complexity, meaning how long it takes to make a prediction when a new ECG is recorded. Table XIV summarizes the time required (in milliseconds) by our models to complete the four main steps of the real-time evaluation: pre-processing (Step 1), extraction of features from groups

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3066347, IEEE Journal of Biomedical and Health Informatics

1,2 and 3 (Step 2), extraction of TSFRESH features (Step 3), model prediction (Step 4). We also provide the 95% Confidence Intervals for each measurement. The Physionet dataset is composed of signals of different length, thus we divide these signals in three groups (less than 20s, between 20 and 40s and more than 40s) in order to have a deeper understanding of how fast the models are with longer recordings. The time complexity of the cross-dataset models is not present in the table because their features are directly coming from the three datasets at the core of our work. Table XIV shows that once a new ECG is available, our model is able to clean it, extract the required features and make a prediction in less than 30 milliseconds for any of the signal present in the data that we analyze (the longest is 61 seconds). This experiment produces sound evidence that our method can be used in a real-time setting. For example, we can comfortably deploy it on a wearable device displaying ECG-based predictions every 50 milliseconds, i.e., 20 times per second.

VI. Limitations

The datasets that we have worked on are proprietary data of external companies, thus we don't have access to its entirety, preventing from making an objective comparison to evaluate directly the performances of our model. However, assuming that the original data is randomly split between training and testing set, meaning that there is no structural difference between the feature characteristics of the two, our Out-of-Sample evaluation provides a reasonable estimate of how the models would perform on the hidden testing sets. As summarized in table XV, it is evident that the models achieve predictive performances that are better or very similar to those from the original papers.

TABLE XV
Performance Comparison (F1-Score).

Paper	Manuscript	Original Work
Physionet [12]	0.93	0.83
Chapman [1]	0.96	0.97
Tianchi [2]	0.99	0.99

For the Physionet model, we report an average F1 Score of 0.93, while the winning model in the 2017 competition achieved only 0.83. For the Chapman data, we achieve average F1 Score of 0.96, while the original paper achieves 0.97. However, in our experiment we keep only one of the 12 ECG leads and we extract only 110 features, so that the overall process is almost instantaneous. For the Tianchi data, we achieve a weighted average $F1$ Score of 0.99, which is in line with the results of the original paper for the labels under consideration. In addition to this, we also propose two experiments in which we train and evaluate models across datasets coming from different sources and having different characteristics, achieving Average F1 Score of 0.99 and 0.93. To our knowledge, there is no other work in the literature that performs a similar analysis.

In an ideal world, having the same set of experts labeling all the ECG signal would be the most accurate and fair way to assess final performance. However, accessing clinical data is particularly challenging, and a cross-country labeling of this kind would be tough to achieve. In this manuscript the set of experts is different in each dataset. While we agree that this issue constitutes a limitation, we also believe that our work, based on the assumptions that labeling standards are equivalent across countries and hospitals, presents sound evidence that this is not far from true. On the contrary, the fact that our models generalize so well on such different populations is a good indicator that the labeling standards are very similar across datasets.

VII. Conclusion

In this manuscript we propose a novel methodology to identify heart anomalies from a newly recorded ECG. The predictive process can be summarized as: signal pre-processing, feature extraction, model training, calibration and evaluation. We design a feature extraction pipeline that crafts 110 features, which we leverage to train five different models on a collection of three datasets. Our models prove to have extremely strong performance when making prediction on unseen data, but are also able to generalize across datasets with ECGs recorded in different settings, and with population having inherently different characteristics. In addition, our approach has showed to be effective for very different kind of heart abnormalities: Normal, Atrial Fibrillation, Tachycardia, Bradycardia, Other (non-specified), Arrhythmia and Noisy. In order to further improve our models' reliability, we calibrate our models using Temperature Scaling to minimize the Expected Calibration Error. Our work confirms that directly analyzing the characteristics of the QRS complex leads to very accurate predictions. This can have an enormous potential impact on the lives of people suffering from heart diseases. In fact, we envisioned our work to be applied in a real time setting, with a wearable device that can constantly monitor the heartbeat of the patients at risk. By designing our experiments to analyze a single lead of a common ECG, we have a good approximation of the input of a given wearable, thus achieving our initial aim without lowering the predictive power of our algorithms. We perform extensive analysis to assess the viability of our models in a real-time setting, and we find that for signals shorter than a minute (the average ECG length is 30 seconds) it takes less than 30 milliseconds from the moment in which the signal is recorded to the final model prediction. As a result, our models prove to be a fast and reliable aid in the important task of detecting heart anomalies from the ECGs of patients who can then be directed to trained experts for further analysis.

Appendix
Features

All the 5 Machine Learning models at the basis of this work are trained on a common set of 110 features extracted according to the procedure described in Section III-D. The extracted features can be divided in four groups: time domain, nonlinear domain, distance based and time series characteristics. The features from the first two groups are extracted through the implementation by Neurokit [23]. Those from the third group are calculated once we detect the P,Q,R,S,T peaks and waves locations in the signal. The time series characteristics are extracted through the TSFRESH implementation [11]. Below we report the complete list of 110 features that are used by the 5 models for their predictions.

A. Time Domain

We select 12 of the initial 13 features for our final model:

- CVNN, The standard deviation of the RR intervals divided by the mean of the RR intervals
- CVSD, The root mean square of the sum of successive differences divided by the mean of the RR intervals
- MCVNN, The median absolute deviation of the RR intervals divided by the median of the absolute differences of their successive differences
- MadNN, The median absolute deviation of the RR intervals
- MeanNN, The mean of the RR intervals
- MedianNN, The median of the absolute values of the successive differences between RR intervals

TABLE XIV
TIME COMPLEXITY ANALYSIS.

Dataset	Length	Pre-Processing	Features Group 1,2,3	Features TSFRESH	Prediction	Total
Physionet	$\leq$ 20s	2.29ms (2.25, 2.34)	16.70ms (16.79, 17.21)	1.05ms (1.03, 1.08)	4.00ms (3.98, 4.01)	24.34ms (24.12, 24.56)
Physionet	20-40s	2.42ms (2.40, 2.44)	19.16ms (19.09, 19.22)	1.50ms (1.49, 1.51)	3.95ms (3.93, 3.96)	27.02ms (26.95, 27.09)
Physionet	$\geq$ 40s	2.47ms (2.46, 2.49)	19.66ms (19.58, 19.74)	1.64ms (1.63, 1.65)	3.95ms (3.94, 3.96)	27.72ms (27.64, 27.81)
Chapman	10s	2.19ms (2.18, 2.20)	17.09ms (17.05, 17.14)	1.00ms (1.00, 1.01)	4.53ms (4.52, 4.55)	24.87ms (24.82, 24.93)
Tianchi	10s	2.32ms (2.31, 2.33)	17.69ms (17.65, 17.72)	0.98ms (0.98, 0.99)	4.00ms (4.00, 4.01)	25.36ms (25.32, 25.40)

- RMSSD, The square root of the mean of the sum of successive differences between adjacent RR intervals
- SDNN, The standard deviation of the RR intervals
- SDSD, The standard deviation of the successive differences between RR intervals
- TINN, An approximation of the RR interval distribution
- pNN20, The proportion of RR intervals greater than 20ms, out of the total number of RR intervals
- pNN50, The proportion of RR intervals greater than 50ms, out of the total number of RR intervals

B. Nonlinear Domain

We select all the initial 7 features for our final model:

- CSI, The Cardiac Sympathetic Index [31]
- CSI Modified, The modified CSI [17]
- CVI, The Cardiac Vagal Index [31]
- SD1, An index of short-term RR interval fluctuations
- SD2, An index of long-term RR interval fluctuations
- SD2SD1, Ratio between short and long term fluctuations of the RR intervals
- SampEn, The sample entropy measure of Heart Rate Variability

C. Distance Based

We select 40 of the initial 109 features for our final model:

- BPM, Beats Per Minute
- IBI, Inter Beat Interval
- Average difference between subsequent R peaks
- Average squared difference between subsequent R peaks
- Average height of R peak
- Median difference between subsequent R peaks
- Median squared difference between subsequent R peaks
- Median height of R peak
- Average, Median, Standard Deviation and Minimum of the horizontal distance between P and Q
- Average, Median and Standard Deviation of the horizontal distance between P and R
- Average, Median and Standard Deviation of the horizontal distance between P and S
- Average, Median and Minimum of the horizontal distance between P and T
- Standard Deviation and Minimum of the horizontal distance between Q and R
- Average of the horizontal distance between Q and T
- Standard Deviation of the horizontal distance between R and S
- Average of the horizontal distance between R and T
- Average, Median and Standard Deviation of the vertical distance between P and R
- Median of the vertical distance between P and T
- Average and Median of the vertical distance between Q and R
- Average of the vertical distance between Q and S
- Average, Median and Minimum of the vertical distance between R and S
- Average, Median, Standard Deviation and Minimum of the vertical distance between R and T

D. Time Series Characteristics

We select 51 of the initial 742 features for our model:

- Abs energy, The absolute energy of the time series
- Agg autocorrelation, Calculates the aggregated variance of the signal
- Agg linear trend (4 sets of parameters), Calculates a linear least-squares regression different aggregated values of the time series
- Augmented dickey fuller, A hypothesis test which checks whether a unit root is present
- Autocorrelation (2 sets of parameters), Calculates the autocorrelation of the signal
- Change quantiles (13 sets of parameters), Fixes a corridor given by the quantiles ql and qh of the distribution of x_v
- Cid ce (2 sets of parameters), An estimate for a time series complexity
- Energy ratio by chunks (2 sets of paramters), Sum of squares of chunks over the whole series.
- Fft aggregated, The spectral centroid (mean) of the absolute fourier transform spectrum
- Fft coefficient (3 sets of parameters), Calculates the fourier coefficients of the one-dimensional discrete Fourier Transform for real inputs
- Friedrich coefficients, Coefficients of polynomial, which has been fitted to the deterministic dynamics of Langevin model
- Index mass quantile, Calculates the relative index i where $q\%$ of the mass of the time series x lie left of i.
- Kurtosis, The adjusted Fisher-Pearson standardized moment coefficient G2
- Large standard deviation, Boolean variable denoting if the standard dev of x is higher than a treshold
- Linear trend attr, Calculates a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one
- Quantile (4 sets of paramters), Calculates the q quantile of x
- Ratio beyond r sigma (8 sets of parameters), Ratio of values that are more than $r * \mathrm{std}(x)$ away from the mean of x.
- Standard deviation, Returns the standard deviation of x
- Time reversal asymmetry statistic
- Variance, Returns the variance of x

RELIABILITY DIAGRAMS

In this section we propose the reliability plots for the calibrated output of each of the five models we train. The reliability plots are calculated through the python implementation [18] of the methodology presented in [15].

A perfectly calibrated plot would have a reliability plot that is exactly corresponding to the 45 degrees line, where accuracy and confidence are equal. In the plots, the dark red gap indicates that the confidence of the model is lower than its accuracy, meaning that the model is under-confident in that bin, while the light red gap indicates the opposite, meaning that it is over-confident. For each model we display a histogram to summarize the representation of each confidence level across 10 bins in the 0-1 probability interval, and the corresponding reliability diagram. Figure 7 presents the reliability plot of the Physionet model, having ECE = 0.035. The model is slightly under-confident in the 0.6-0.9 interval and is over-confident in the 0.5-0.6 one. Again, this is due to the small size of the test set of this model (300 samples).

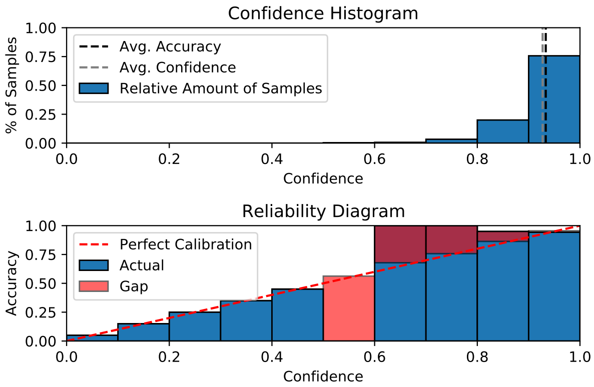

Fig. 7. Reliability Diagram Physionet.

Figure 8 presents the reliability plot of the Chapman model, having ECE = 0.006. This model is almost perfectly calibrated, and is only slightly over-confident in the 0.4-0.6 and 0.7-0.9 intervals.

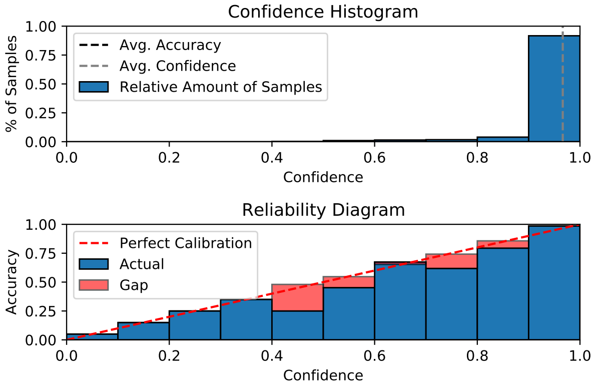

Fig. 8. Reliability Diagram Chapman.

Figure 9 presents the reliability plot of the Tianchi model, having ECE = 0.001. Also this model has almost perfect calibration and is slightly under-confident in the 0.3-0.4 interval.

Figure 10 presents the reliability plot of the model trained on Chapman and evaluated on Tianchi, having ECE = 0.0008. This model has the lowest calibration error and is basically perfect in its predictions.

Figure 11 presents the reliability plot of the model trained on Physionet and evaluated on Chapman, having ECE = 0.02. Despite being the most challenging model to train, as the training and testing datasets come from different countries, hospitals and have different recording standards, this model is still remarkably well calibrated.

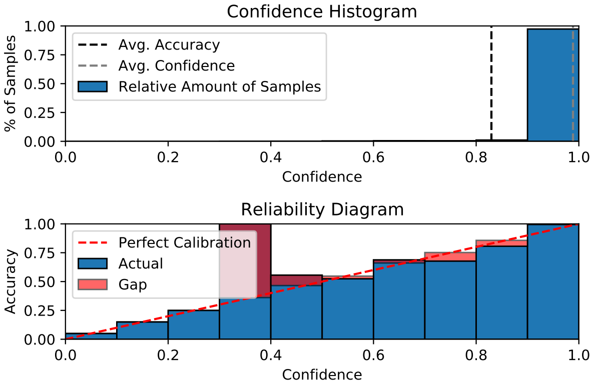

Fig. 9. Reliability Diagram Tianchi.

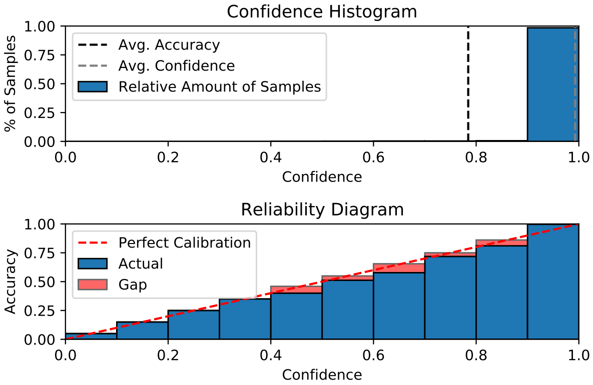

Fig. 10. Reliability Diagram Chapman-Tianchi.

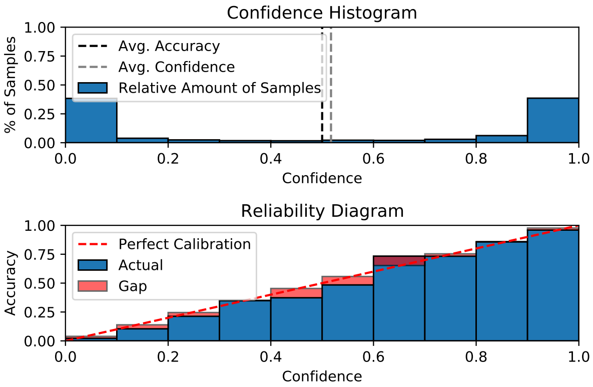

Fig. 11. Reliability Diagram Physionet-Chapman.

References

[1] Chapman university and shaoxing people's hospital. https://figshare.com/collections/ChapmanECG/4560497/1, 2019.

[2] Tianchi hefei high-tech cup ecg human-machine intelligence competition. http://tianchi-competition.oss-cn-hangzhou.aliyuncs.com/231754/round2/hf_round2_train.zip, 2019.

[3] Alivecor, Inc. https://www.alivecor.com/#, 2020.

[4] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[5] Z. D. G. Ary L. Goldberger and A. Shvilkin. Goldberger's clinical electrocardiography. https://www.sciencedirect.com/topics/medicine-and-dentistry/qrs-complex, 2017.

[6] Z. I. Attia, P. A. Noseworthy, F. Lopez-Jimenez, S. J. Asirvatham, A. J. Deshmukh, B. J. Gersh, R. E. Carter, X. Yao, A. A. Rabinstein, B. J. Erickson, et al. An artificial intelligence-enabled ecg algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction. *The Lancet*, 394(10201):861–867, 2019.

[7] S. Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[8] J. Cai, W. Sun, J. Guan, and I. You. Multi-ecgnet for ecg arrythmia multi-label classification. *IEEE Access*, 8:110848–110858, 2020.

[9] G. A. Campbell. Electric wave-filter., May 22 1917. US Patent 1,227,113.

[10] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[11] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (TSFRESH–a python package). *Neurocomputing*, 307:72–77, 2018.

[12] G. D. Clifford, C. Liu, B. Moody, H. L. Li-wei, I. Silva, Q. Li, A. Johnson, and R. G. Mark. Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.

[13] C. for Disease Control and Prevention. Deaths and mortality. https://www.cdc.gov/nchs/fastats/deaths.htm, May 2017.

[14] C. for Disease Control and Prevention. Atrial fibrillation. https://www.cdc.gov/heartdisease/atrial_fibrillation.htm, May 2020.

[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[16] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65, 2019.

[17] J. Jeppesen, S. Beniczky, P. Johansen, P. Sidenius, and A. Fuglsang-Frederiksen. Using lorenz plot and cardiac sympathetic index of heart rate variability for detecting seizures for patients with epilepsy. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4563–4566. IEEE, 2014.

[18] F. Küppers, J. Kronenberger, A. Shantia, and A. Haselhoff. Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[19] G. Y. Lip, P. Kakar, and T. Watson. Atrial fibrillation—the growing epidemic. *Heart*, 93(5):542, 2007.

[20] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[21] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[22] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[23] D. Makowski. Neurokit: A python toolbox for statistics and neurophysiological signal processing (eeg eda ecg emg...). *Memory and Cognition Lab'Day*, 1, 2016.

[24] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[25] M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[26] S. L. Oh, E. Y. Ng, R. San Tan, and U. R. Acharya. Automated diagnosis of arrhythmia using combination of cnn and lstm techniques with variable length heart beats. *Computers in biology and medicine*, 102:278–287, 2018.

[27] J. Pan and W. J. Tompkins. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME–32(3):230–236, 1985.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] M. V. Perez, K. W. Mahaffey, H. Hedlin, J. S. Rumsfeld, A. Garcia, T. Ferris, V. Balasubramanian, A. M. Russo, A. Rajmane, L. Cheung, et al. Large-scale assessment of a smartwatch to identify atrial fibrillation. *New England Journal of Medicine*, 381(20):1909–1917, 2019.

[30] L. Sathyapriya, L. Murali, and T. Manigandan. Analysis and detection R-peak detection using modified pan-tompkins algorithm. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pages 483–487. IEEE, 2014.

[31] M. Toichi, T. Sugiura, T. Murai, and A. Sengoku. A new method of assessing cardiac autonomic function and its comparison with spectral analysis and coefficient of variation of r–r interval. *Journal of the autonomic nervous system*, 62(1-2):79–84, 1997.

[32] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[33] J. Zheng, H. Chu, D. Struppa, J. Zhang, M. Yacoub, H. El-Askary, A. Chang, L. Ehwerhemuepha, I. Abudayyeh, A. Barrett, et al. Optimal multi-stage arrhythmia classification approach. *Scientific reports*, 10(1):1–17, 2020.