

# ECG Classification Based on Time and Frequency Domain Features Using Random Forests

Martin Kropf<sup>1,2,3</sup>, Dieter Hayn<sup>2</sup>, Günter Schreier<sup>2,3</sup>

<sup>1</sup>Charité Virchow-Klinikum, Berlin, Germany

<sup>2</sup>AIT Austrian Institute of Technology, Graz, Austria

<sup>3</sup>TU Graz, Graz, Austria

## Abstract

*We present a combined method of classical signal analysis and machine learning algorithms for the automated classification of 1-lead ECG recordings, which was developed in the course of the Computing in Cardiology Challenge 2017.*

*To classify ECG recordings into the four classes as defined for the Challenge (normal, suspicious to AF, suspicious to other arrhythmia, noise) we used MATLAB and a set of algorithms for detection of beats, wave point detection on detected beats, quality evaluation of the detection, averaging of beats, beat classification, rhythm classification and many more. We extracted a variety of features from both time and frequency domain etc. as input features for the classifier. A total of 380 features were used to train a Random Forest –based classifier (bagged decision trees). Since classes for the Challenge were severely unbalanced, weights based on the class distribution were applied. To train the classifier and for our internal evaluation we used cross-validation on all available ECGs from the training-set.*

*10-fold cross-validated F1 score on the training set is 0.83. Final F1 score from the official challenge evaluation on the enhanced dataset is 0.81, which is quite close to the other top performing algorithms.*

## 1. Introduction

Recently developed smartphone-based ECG recorders enable screening for arrhythmias in a broad population. However, for successful application of such approaches, algorithms specialized for single lead ECGs are required that provide higher reliability than state of the art ECG processing tools if applied to just a single ECG channel. We present a combined method of classical signal analysis and machine learning algorithms for the automated classification of 1-lead ECG recordings, which was developed in the course of the Computing in Cardiology Challenge 2017.

## 1.1. AF detectors

Several approaches to detect AF have been described in the literature. Classical signal analysis approaches rely either on deriving atrial activity by means of P wave analysis or by investigating the ventricular response, i.e. calculating regularity of RR intervals. Recently, machine learning algorithms have emerged for the detection of AF. Several different approaches have been described: decision trees [1], neural networks [2,3] and support vector machines [4]. A recent publication even claims to reach cardiologist-level accuracy in classifying ECGs collected on a mobile device based on deep learning [3].

In this work we present a combination of analyzing atrial activity and ventricular response and feed the extracted input features in a Random Forest-based classifier.

## 2. Methods

For processing the ECGs we used MATLAB 2017a (The MathWorks, Natick, MA, USA) and several custom tools, which are described further in the following sections.

### 2.1. Datasets

The algorithm described has been developed using the AF Classification challenge 2017 database from PhysioNet [4].

Data for the challenge consisted of a collection of 8.528 recordings, lasting from 9s to 60s. Each recording included one noninvasively ECG signal which was obtained using a mobile ECG recording device (AliveCor® KardiaMobile) with 300 samples per second.

Challenge data comprised two different sets: a training set (8.528 ECGs, reference annotations for participant's use available), and a hidden test set (3.658 ECGs, unpublished records, reference annotations withheld) [4]

## 2.2. ECG preprocessing

We used several algorithms of our existing ECG Signal Processing Toolbox (which is described in detail in [5]) for preprocessing of the ECG signals.

**QRS detection** was done with two different QRS detectors:

1. pre-existing QRS detector [5]
2. CinC 2017 sample entry QRS detector [6]

**Lead inversion detection** by analyzing the signs of P, QRS and T wave [34]. If a lead was detected to be inverted, we flipped the ECG by multiplying the signal with  $-1$  and started the preprocessing step again.

**Average beat calculation / beat classification** based on a predefined correlation threshold, classes of averaged beats with similar morphology (i.e. high correlation coefficient) were generated.

**Rhythm classification** was performed by aggregating the classification results of single beats. Subsequently, RR-interval based features, e.g. atrial heart rate or ventricular heart rate, were calculated.

**Detect wavepoints:** P and T waves were detected by analyzing peaks in predefined windows around the QRS complex and features like amplitudes, distances and durations were calculated.

**Atrial signal analysis:** after removing ventricular components (QRS) from the signal we searched for peaks in the remaining signal and calculated atrial heart rate and regularity.

After preprocessing the ECG feature, fusion was performed by combining the weighted sum of individual parameters. We created several combined features, e.g.:

**Normal or Noise:** Combination of QRS detection quality measure [7], the GINI coefficient of QRS amplitudes, GINI coefficient of QRS amplitudes and the number of beats for primary class.

**Normal or AF:** Combination of RR irregularity, P-amplitude, Atrial Heart Rate, Percent of premature beats

**AF or Other:** Combination of RR irregularity, P-amplitude, BCI skewness, number of samples, number of premature beats and number of ectopic beats.

## 2.6. Modelling

We used our existing MATLAB Predictive Modelling Pipeline [8] to test multiple **Random Forest** (RF) configurations on the training data set. The RF algorithms form a family of classification methods that rely on the combination of several decision trees (Figure 2). [9].

**10-fold cross validation** was used to avoid over-fitting. 10 different sub-models were build on 10 training sets

containing 90 % of the whole data. The remaining 10 % were used each model. Figure 1 shows the classification workflow.

Figure 1: ECG classification workflow

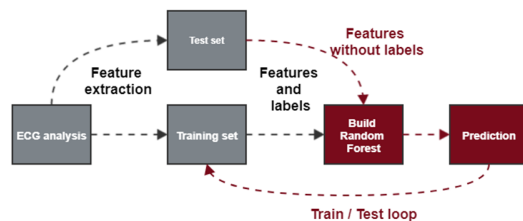
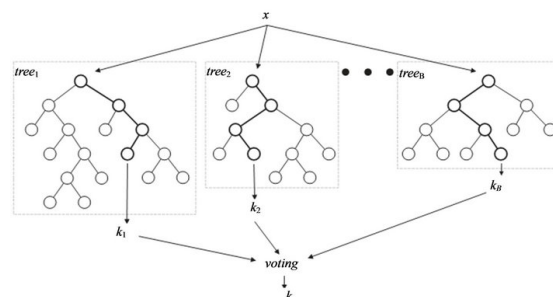


Figure 2: General architecture of a Random Forest



A **template tree** has been constructed, which parameters are described in Table 1.

Because the number of instances for the classes were severely unbalanced, **observation weights** and **misclassification costs** were passed to the RF classifier. Weights and costs have been calculated based on the number of instances for each class in the training data set.

Table 1: Configuration of template tree for Random Forest classifier.

Parameter	Value
Surrogate	2
MaxNumSplits	10000
NumVariablesToSample	30
MergeLeaves	Off
MinLeaf	1
Prune	On
PruneCriterion	Impurity
SplitCriterion	Deviance
PredictorSelection	AllSplits
AlgorithmForCategorical	Exact
MaxNumCategories	10
QuadraticErrorTolerance	$10^{-4}$

Figure 3: Predictive Modelling Toolbox ECG viewer (Training set signal A00077)

1: unfiltered signal, 2: RR intervals and beat classification, 3: averaged beat view for four classes (including wavepoint detection), 4: feature table comparing feature values of single signals to mean value of true positive classified signals



## 2.7. Visual signal, feature and classification results evaluation

For evaluating our classification results we implemented a MATLAB-based graphical user interface (GUI) for ECG viewing (see Figure 3) into our existing Predictive Modelling Pipeline [8]. The GUI can be opened directly by clicking on a field in the confusion matrix to display signals based on the classification result (e.g. show only signals of reference class *Other* which have been erroneously predicted as *AF*). Figure 3-1 shows the signal itself. Figure 3-2 shows the detected R-peaks, arranged by beat-coupling-interval, with their detected classes.

Figure 3-3 shows up to four averaged beat classes (based on morphology) with their detected P and QRS start and end points. Figure 3-4 shows the calculated features for the current signal in a table view and provides a comparison of the current signal features to the calculated mean values of the true positive classified signals. Features can either be sorted alphabetically or according to their Feature Influence on the classification result for the current signal as presented in detail in [10]. The GUI supports loading of different processing configurations and classification results.

## 2.8 Evaluation

For evaluation we calculated different key performance indicators, including the tailored F1 score according to the challenge requirements as described in Clifford et al. [6]

## 3. Results

We have tested several configurations of the RF-classifier and provide the detailed results on the training data set in Table 2.

Table 2. Unofficial F1 scores for 10-fold CV on training set

Class	100 trees, no weights	100 trees, weights	1000 trees, no weights	1000 trees, weights
Normal	0.891	0.898	0.891	0.898
AF	0.773	0.825	0.775	0.830
Other	0.734	0.757	0.735	0.760
Noise	0.212	0.641	0.185	0.648
Total	0.799	0.827	0.800	0.830

Our code which gained best results in our internal tests on the training data was submitted to the automated scoring system and evaluated against the test set, as described in [6]. The entry was evaluated by the automatic scoring system and F1 score for every class was calculated.

Table 3. Official F1 scores provided by challenge scoring system

Class	Phase 1	Phase 2
Normal	0.84	0.90
AF	0.79	0.83
Other	0.75	0.72
Noise	0.44	-
Total	0.71	0.82

Finally, our algorithm was applied by the challenge organizers on the extended, hidden test set [6]. Provided scores for the test-set were 0.81 (Rank 9), for training 0.9349

#### 4. Conclusion

Our approach works with classical machine learning algorithms using moderate computational resources and physiologically motivated feature engineering, thus reducing the risk of severe over-fitting. The performance compares well to the top scores of other challenge participants.

We did not refine the reference annotations yet, although we noticed that some of the annotations were contradicting. Refinement of those annotations could probably lead to better classification performance. We plan to perform tests with refined reference annotations and to augment the existing dataset to perform additional tests of our algorithm with other databases.

#### Acknowledgements

Thanks to Prof. Daniel Scherr, MD, and David Zweiker, MD, from the Department of Cardiology of the Medical University of Graz, who helped us to understand some unclear ECG classifications.

#### References

- [1] Charfi F, Kraiem A. Comparative Study of ECG Classification Performance Using Decision Tree Algorithms. *Int J E-Health Med Commun.* 2012;3(December):102–20.
- [2] Wang TF, Devine B, Macfarlane PW. Artificial neural networks for the diagnosis of atrial fibrillation. *Med Biol Eng Comput.* 1994;32(6):615–9.
- [3] Rajpurkar P, Hannun AY, Haghpanahi M, Bourn C, Ng AY. Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks. 2017; Available from: <http://arxiv.org/abs/1707.01836>
- [4] Colloca R, Johnson AEW, Mainardi L, Clifford GD. A Support Vector Machine approach for reliable detection of atrial fibrillation events. *Comput Cardiol Conf (CinC), 2013.* 2013;1047–50.
- [5] Hayn D. Development, Validation, and Optimisation of High-Quality ECG Processing Algorithms. 2007.
- [6] Clifford G, Liu C, Moody B, Silva I, Li Q, Johnson A, et al. AF Classification from a Short Single Lead ECG Recording: the PhysioNet Computing in Cardiology Challenge 2017. *Comput Cardiol (Rennes IEEE).* 2017;44.
- [7] Hayn D, Jammerbund B, Schreier G. ECG Quality Assessment for Patient Empowerment in mHealth Applications Introduction Criterion A1 – Signal amplitude Criterion A2 – Spike detection Criterion A4 – Total length of remaining signal Criterion B – Number of lead crossing points Criterion D – Q. *Comput Cardiol.* 2011;38:353–6.
- [8] Hayn D, Kreiner K, Ebner H, Kastner P, Breznik N, Rzepka A, et al. Development of Multivariable Models to Predict and Benchmark Transfusion in Elective Surgery Supporting Patient Blood Management. *Appl Clin Inform.* 2017;8(2):617–31.
- [9] Nguyen C, Wang Y, Nguyen HN. Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *J Biomed Sci Eng [Internet].* 2013;6(5):551–60. Available from: <http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/jbise.2013.65070>
- [10] Hayn D, Walch H, Stieg J, Kreiner K, Ebner H, Schreier G. Plausibility of individual decisions from random forests in clinical predictive modelling applications. *Stud Health Technol Inform.* 2017;236:328–35.

Address for correspondence.

Martin Kropf  
 Fuldastraße 59  
 12043 Berlin  
 Germany  
[Martin.Kropf@charite.de](mailto:Martin.Kropf@charite.de)