2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems

A Machine Learning Approach to Objective Cardiac Event Detection

N. Twomey*, P. A. Flach†
Department of Engineering* and Computer Science†, University of Bristol, United Kingdom
niall.twomey@bristol.ac.uk

***Abstract*—This paper presents an automated framework for the detection of the QRS complex from Electrocardiogram (ECG) signals. We introduce an artefact-tolerant pre-processing algorithm which emphasises a number of characteristics of the ECG that are representative of the QRS complex. With this processed ECG signal we train Logistic Regression and Support Vector Machine classification models. With our approach we obtain over 99.7% detection sensitivity and precision on the MIT-BIH database without using supplementary de-noising or pre-emphasis filters.**

***Index Terms*—Pattern recognition, QRS detection**

I. Introduction

Researchers have investigated algorithms to accurately and consistently identify cardiac events for over forty years. The ECG, as a measure of the electrical stimulation of the heart, provides a means to measure one of the most important characteristics of cardiac activity — the heart beat (Figure 1). A heart beat can be characterised by five deflections in the ECG which are termed the P, Q, R, S and T waves. The region between the Q and S waves is known as the QRS complex and is generally the most pronounced feature of the ECG. Discovery of the QRS complex is a preliminary requirement for the assertion and quantification of a number of important heart rate variability metrics [1] and for automatically identifying the less pronounced P and T waves.

The standard means of detecting these QRS complexes has been to apply linear and non-linear filters to raw ECG streams and then to compare the output of these filters to sets of thresholds that have been heuristically chosen by the researchers. Many algorithms also employ elaborate post-processing in order to select and reject candidate QRS complexes so their frequency is constrained to be within biologically feasible bounds. Without such post-processing techniques many of these filtering approaches often yield poor detection accuracy.

One shortcoming of this standard approach is that the researchers must expend a great deal of time and effort to select thresholds that generalise well, and there is no guarantee that the chosen parameters are either calibrated or optimal in any sense. Indeed, a number of the parameters that are chosen are dependent on the age of the individual which indicates that these hand-crafted algorithms should only be applied to a particular age demographic. Reframing these filtering algorithms to accommodate varying demographics might necessitate the full redefinition of all parameters.

One final concern about the standard approach is that few research groups have employed cross-validation during performance evaluation or during parameter selection. This is an important point and indicates that many of the results in such publications may be optimistically biased.

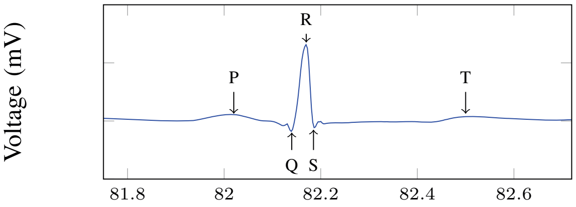

Figure 1: An example ECG trace.

In this work, we employ a machine learning approach to QRS detection using the MIT Beth Israel Hospital (MIT-BIH) [2] arrhythmia database; a standard ECG database upon which many QRS detection algorithms are benchmarked. Using machine learning techniques enriches the affordances of QRS detection as, in particular, computer-based parameter selection can search very high-dimensional feature spaces. Additionally, this approach is capable of utilising such feature spaces to obtain calibrated decision boundaries that are optimal with regard to particular operating characteristics.

Our experimental results are promising and we will show that accurate QRS detection can be systematically obtained from single-lead ECG signals. The contributions of this paper include — 1) Our technique automatically obtains good tolerance to signal artefacts, arrhythmia, ectopia, and other anomalies that obstruct perfect QRS detection. 2) Our approach returns very high accuracy when using only basic pre- and post-processing. 3) Our approach provides a grounded means of specifying QRS points from a stream of ECG data, as, when multiple candidates are available, we select the points which yield the highest *probability* of being the true QRS point. 4) We also present a general QRS complex detection framework that can learn models for varying demographics.

II. Methods

A. Data

In this paper, we investigate the MIT-BIH arrhythmia database [2]. This database consists of 30-minute, two-channel ambulatory ECG recordings, which were recorded between 1975 and 1979. Twice-validated expert annotations accompany this database and these identify QRS and non-QRS periods.

978-1-4799-4325-8/14 $31.00 © 2014 Crown Copyright
DOI 10.1109/CISIS.2014.75

Regions of artefact contamination are further identified. We do not consider these times. The ECG is sampled at 360 Hz. We assess the performance of our approach on the ECG channel containing fewer signal artefacts.

B. Pre-processing

The intuition behind our pre-processing methodology will first be introduced with reference to the idealised ECG trace shown in Figure 1. In Section III we will show that this algorithm is also capable of adapting to non-ideal ECG. In Figure 1 one can observe a sharp positive deflection in the ECG between the Q and R waves, and similarly a sharp negative deflection between the R and S waves. No such pattern is seen elsewhere in this ECG segment. The goal of the proposed method is to provide a measure of these positive and negative slopes in a manner that can be understood by a learning algorithm.

The length of the QRS complex has been stated in a number of studies as being $\approx$ 88 ms [3] (approximately 32 samples at 360 Hz). Therefore, we employ a sliding window over the ECG of length of 32 samples. If a QRS complex is centred in this window (i.e. the R wave in Figure 1 is in the centre of the window), the Q wave should be found in the left hand side of the sliding window, and the S wave should be found in the right hand side of the window. By computing the least square slopes on the left and right hand halves of the sliding window independently we measure the degree to which segments of the ECG may resemble a QRS complex.

The final step in pre-processing is to compute the negative product of the least square slopes from the left and right hand sides of the sliding window. Observing steep positive slopes on the left hand side and steep negative slopes on the right hand side are indicative of a QRS complex. When outside of a QRS complex, smaller slopes will be measured which will suppress non-QRS shapes in the processed ECG. Therefore, computing this product will emphasise the QRS segments and suppress the representation of non-QRS segments. Furthermore, taking the product of these slopes is a requirement to allow linear classifiers to solve the problem (see Section III).

We demonstrate this algorithm in Figure 2. Figure 2a shows the ECG signal trace (blue) and the sliding window localised at these three periods in time (grey). Figure 2b has split each sliding window into the left and right sections (red and green), and it also shows the least square regression lines that are computed for the ECG within these regions (see the locations marked $m_{1,l}$ — $m_{3,r}$). Finally, Figure 2c shows the negative product of the least square slopes from the left and right hand side of the sliding window. This example shows that the QRS complex exemplifies the most significant deflections in the ECG signal trace and this is also seen in Figure 2c.

We provide three simple feature to the learning algorithms: the processed ECG signal and its first and second derivative.

C. Learning Algorithms

We investigate the performance of Logistic Regression (LR) and Support Vector Machines (SVMs) on QRS detection. The reason for assessing these two algorithms is because LR is a simple algorithm that learns a decision boundary directly using only the input features alone, whereas SVMs are capable of increasing the dimensionality of the feature space to assist with the classification problem. Assessment of both algorithms will shed light on the complexity requirements of the task.

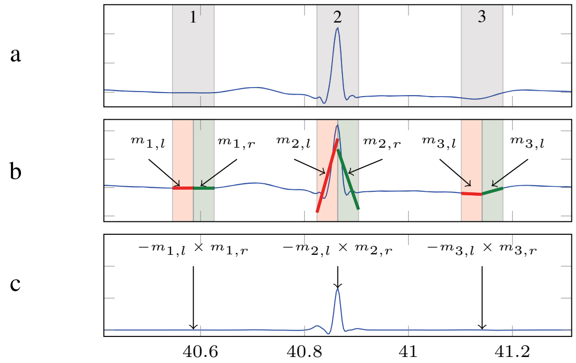

Figure 2: Visual demonstration of pre-processing algorithm.

1) Logistic Regression: LR is one of the most commonly used discriminative probabilistic models. Assuming an input vector, $\boldsymbol{x}$ (length n) a LR model learns a weight vector, $\boldsymbol{w}$ (length n) and a bias term, b. Decisions are made by testing the value of $\boldsymbol{w} \cdot \boldsymbol{x} + b$ after learning their values from data. We condense our notation by assuming $x_0 = 1$ and that $w_0 = b$ so that $\sum_{i=1}^{n} \boldsymbol{w}_i \cdot \boldsymbol{x}_i + b \equiv \sum_{i=0}^{n} \boldsymbol{w}_i \cdot \boldsymbol{x}_i \equiv \boldsymbol{w} \cdot \boldsymbol{x}$.

The conditional probability of the class y is computed with the sigmoid function

$$\widehat{p} = p(y|\boldsymbol{x};\boldsymbol{w}) = \frac{1}{1 + \exp\left(-\boldsymbol{w} \cdot \boldsymbol{x}\right)}. \tag{1}$$

Assuming that $y = 1$ for positive examples (i.e. QRS points), and $y = 0$ for negative examples, each example from a dataset defines a Bernoulli distribution according to Equation (1). The Log Conditional Likelihood (LCL) of the training data (of length m) is computed with

$$\mathrm{LCL}(\boldsymbol{w}) = \sum_{i-1}^{m} y_i \log \widehat{p} + (1 - y_i) \log\left(1 - \widehat{p}\right), \tag{2}$$

and the optimal weights, $\widehat{\boldsymbol{w}}$, are those that maximise this

$$\widehat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \mathrm{LCL}(\boldsymbol{w}). \tag{3}$$

$\widehat{\boldsymbol{w}}$ is learnt with algorithms such as gradient descent which iteratively update these weights to maximise the LCL. Specification of $\widehat{\boldsymbol{w}}$ defines a linear decision boundary in feature space for a given probability threshold (e.g. 0.5). In general, models with large weights tend to overfit the training data, so we penalise these and get

$$\widehat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \mathrm{LCL}(\boldsymbol{w}) - \frac{\lambda}{2} \left\| w \right\|_2^2, \tag{4}$$

where λ is known as the regularisation parameter and is selected by the user. Equation (4) defines a convex optimisation problem, so the gradient of the data is calculated with respect to the weighs. The weights are then iteratively updated. Applying the gradient descent update rule specifies that the k^{th} weight will assume the following value after j iterations

$$\boldsymbol{w}_k^{(j)} = \boldsymbol{w}_k^{(j-1)} + \eta \left(\sum_{i=1}^{m} (y_i - \widehat{p}) \boldsymbol{x}_k - \lambda \boldsymbol{w}_k^{(j-1)} \right), \tag{5}$$

where the value η is termed the 'learning rate' and defines the size of the step that is taken in the direction of the gradient. The initial weights, $\boldsymbol{w}^{(0)}$, are randomly initialised. Weight updating is repeated until changes in $\boldsymbol{w}$ have negligible effect on the LCL.

2) Support Vector Machines: SVMs [4] are another popular type of discriminative model. SVMs select the weight parameters that maximise the margin of classification using the hinge loss function [4]. We employ the condensed notation as before, i.e. $x_0 = 1$ and $w_0 = b$, but also change the value of labels for the negative class to $y = -1$. The following then specifies the constrained optimisation problem for linear SVMs

$$\begin{aligned} \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\zeta}} = & \arg\min_{\boldsymbol{w},\boldsymbol{\zeta}} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{m} \boldsymbol{\zeta}_i \\ & \text{subject to } y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i \right) \geq 1 - \boldsymbol{\zeta}_i \text{ and } \boldsymbol{\zeta}_i \geq 0, \ \forall i \end{aligned} \tag{6}$$

where $\boldsymbol{\zeta}_i$ are known as slack variables and allow the SVM to learn a maximum margin decision boundary when the data are not linearly separable. C is known as the complexity parameter and performs a similar role as λ in LR.

SVMs can convert classification tasks into higher dimensions with the use of kernel functions [5]. We employ the Radial Basis Function (RBF) kernel (Equation (7)) in this work which takes an input example, $\boldsymbol{x}$, a support vector, $\boldsymbol{x}'$, and a scaling parameter, γ, as arguments. Support vectors are the training examples which were incorrectly classified or lay near the decision boundary during training.

$$K_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{x}'; \gamma) = \exp\left(-\gamma \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|_2^2 \right). \tag{7}$$

$\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$ is nothing more than the squared Euclidean distance between the test example and the support vector, and so the RBF kernel can be viewed as a 'similarity' measure, as small distances yield values close to 1 whereas large distances yield values closer to 0. γ, sometimes known as the bandwidth parameter, must be positive and specifies the influence that a particular support vector has in its locality in feature space. Smaller γ will push the value towards 1, while larger γ will draw the output of the kernel towards 0.

The optimisation problem in Equation (6) can be solved with Lagrangian multipliers, and SVM models are trained on computers with quadratic optimisation routines, such as LibSVM [6]. The output of the SVM is the distance to the decision surface which can be converted to probabilistic outputs using calibration methods described in [7].

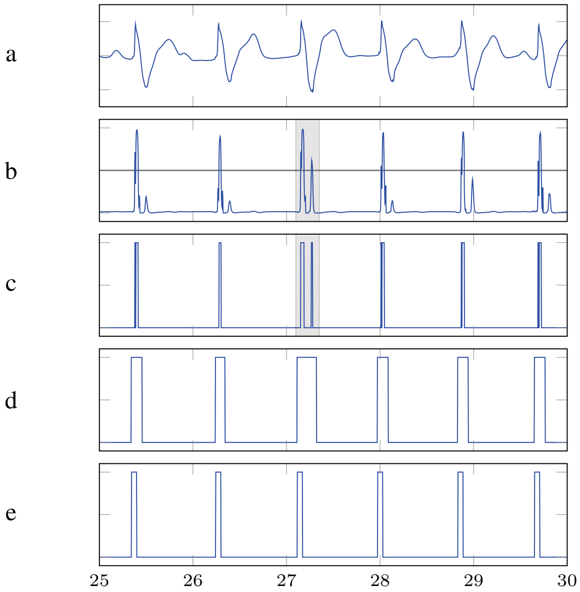

Figure 3: The post-processing algorithm outputs.

D. Parameter Selection

We have already mentioned a number of parameters that must be selected (λ for LR models, and (C, γ) for SVMs). In many research papers, classification hyperparameters are selected after a coarse-to-fine grid search which iteratively 'zoom' in on the best performing parameters for a fixed number of steps or until termination criteria are satisfied.

In our work, however, instead of providing a set of parameters, we provide a *distribution* of possible parameters. During the selection routine, the algorithm randomly samples from this distribution and assesses the performance at these points. This approach provides a number of advantages over a grid search. Most importantly, if one hyperparameter is unimportant, a traditional grid search investigates many unimportant values while keeping important parameters constant, and therefore performs many redundant queries. However, when sampling from a distribution, this is not the case and a greater assessment of the landscape of the optimisation objective can be obtained after the same number of iterations [8].

We perform five-fold cross validation on the training data. Test data are never seen by the parameter selection algorithm.

E. Post-processing

We employ simple post-processing techniques in this work in comparison to other approaches [9]. Figure 3 shows an example of the post-processing methodology with unhealthy ECG. The unhealthy ECG can be identified by noting that the ECG in Figure 3a is dissimilar to the healthy ECG shown in Figure 1. It is important to note that with unhealthy ECG, it is often difficult to even assign P, Q, R, S and T labels visually and algorithmically. Subsequently we must perform post-processing on the classifier's initial decisions.

In Figure 3b, we show the QRS probability as given by a LR classifier. If these probabilities are thresholded at 0.5, we obtain a set of *plausible* QRS regions that are shown with values of 1 in Figure 3c. In some cases multiple plausible regions are nominated from a single QRS complex, e.g. the highlighted regions in Figures 3b, and these mostly occur with unhealthy patients' ECG.

Our post-processing algorithm 'stretches' these plausible regions over time so that those in close proximity are absorbed into one another (Figure 3d). After this, the predicted location of the QRS complex has become more ambiguous because the plausible regions now occupy a longer duration of time. Recall that we assume here that the QRS complex lasts for 88 ms, so we must return a more specific decision. To do so, we consider all possible sequences of probabilities of length 88 ms within the jurisdiction of the stretched window. The sequence that produces the largest sum is nominated by the algorithm as the most likely candidate. This results in the narrower windows in Figure 3e.

F. Performance Evaluation

We employ Leave One Out (LOO) for performance evaluation as our database is relatively small and because LOO is known to be an almost unbiased estimator of true generalisation error [10]. With LOO, data from all but one patient are used to learn the model parameters. The resulting model is then tested against the held-out data.

We assess the performance of our classifiers with the *sensitivity* and *precision* metrics. Sensitivity measures the true positive classification rate and precision measures the proportion of true positives among the predicted positives. These are computed by

$$\text{Sensitivity} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}}, \text{ and} \tag{8}$$

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FP}}. \tag{9}$$

We also compute the harmonic mean of these (F_1 score).

We compare the average rank of the classifiers. Performance assessment metrics are calculated over the whole database, and for each patient the classifier that obtained in the best score is given the rank 1, and the other is given the rank 2. Calculating the average rank of classifiers identifies the degree to which one classifier outperforms the other. Average ranks closer to 1 indicate that one classifier nearly always outperforms the other on a particular metric.

III. Results and Discussion

A. QRS Detection

Table I tabulates the sensitivity, precision and F_1 scores that were obtained from the 24 patients of the MIT-BIH database. We obtained over 99.7% sensitivity, precision and F_1 scores on average with both LR and SVM models. These are very high results and indicate that our approach is well suited for QRS detection. Indeed, our results are competitive, and sometimes better, to those obtained by many other researchers in this area [1] even having used much simpler pre- and post-processing algorithms. In particular our approach yields a consistently lower gap between the sensitivity and precision performance where many researchers obtain differences of $\approx 1\%$. Indeed, these results were obtained in an objective framework where testing data were never seen in training. This is promising as it suggests that our results may be improved even further by adopting more elaborate procedures on top of those we discussed earlier.

Table I: QRS detection results.

	LR			SVM		
	Sens.	Prec.	F_1	Sens.	Prec.	F_1
Mean	0.997	0.999	0.998	0.997	0.999	0.998
Std. Dev.	0.004	0.000	0.002	0.004	0.001	0.002
Min	0.978	0.996	0.989	0.981	0.995	0.991

Table II: Average rank of classifiers.

	LR Rank	SVM Rank
Sensitivity	1.17	1.22
Precision	1.00	1.17
F_1 Score	1.13	1.30

One of the most striking results is that the mean sensitivity, precision and F_1 scores from LR and SVM models are very similar. The average ranks of classifiers are shown in Table II. This table shows that the best rank is obtained by LR models over all performance assessment metrics. This strongly suggests that LR is the more suitable model for classification.

These results are somewhat surprising as the simpler model seems to perform better on all performance assessment metrics. There are a number of possible explanations for this. Firstly, due to the complex nature of training SVMs, we provided them with 100,000 randomly selected training examples. LR, however, uses a simpler optimiser which does not expand the feature space during training, and therefore the final weights were learnt from approximately 1,000,000 labelled training examples. It is possible that the main strength of the LR model is as a result of the larger quantity of training data afforded to it.

It is worth noting, however, that only a few simple features were used to perform classification in this work. Achieving results that are competitive with the state of the art with such simple classifiers suggests that our pre-processing algorithm is very well suited to the QRS detection problem. We expect that extracting more descriptive features from both the raw and processed ECG signals would increase the overall sensitivity and precision of our algorithm.

[1] e.g. [9] obtained $\approx 99.6\%$ sensitivity and precision; [11] obtained 99.64% sensitivity and 99.82% precision; and [12] obtained 99.71% and 99.53% sensitivity and precision.

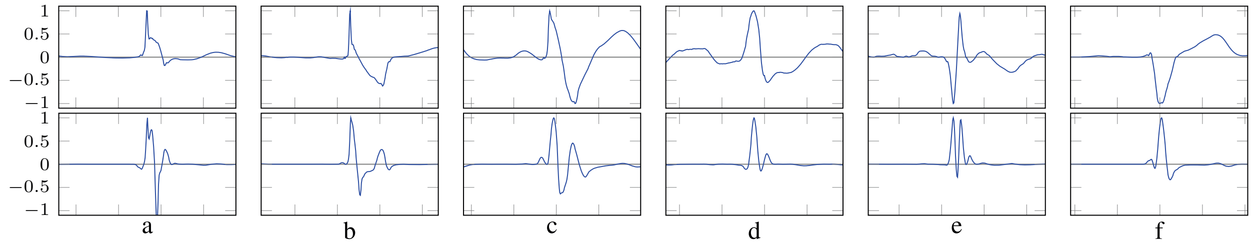

Figure 4: Adaptation of the pre-processing algorithm to abnormal ECG signals.

B. Pre-processing

1) Representation: Figure 4 shows the outputs of our pre-processing algorithm on various categories of non-normal ECG. This image shows a grid of six ECG signal traces (top row) and the six outputs of the pre-processing algorithm (bottom row) during the same time period.

Our pre-processing algorithm boasts a number of interesting features. Firstly, when considering a clear peak in the ECG, the pre-processing algorithm is guaranteed to output a *rectified* version of the peak. This is because a peak can be loosely described as consisting of a positive and negative slope centred on an apex. Taking the negative product of these slopes will always yield a positive number. In other words, after pre-processing, all QRS complexes should be found in the positive half of the processed signal space. This is exemplified in Figures 4e and 4f where the negative peaks (top) in the ECG are rectified after the processing algorithm (bottom).

Similarly, sustained rising and falling trends are guaranteed to yield negative values after pre-processing. This is because the signs of both slopes on the left and right halves of sliding window are the same, and their negative product must itself be negative. As QRS complexes are defined by a peak, we can state that QRS complexes should not be found in the negative half of this space. Therefore only non-QRS waves will be found in this region. Figures 4a, 4b and 4c show this.

Finally, ‘slow’ waves (Figure 4d) are also significantly reduced in amplitude after pre-processing. This is due to the fact that the slope products during these periods are close to zero or are negative. We demonstrate this explicitly in Figure 5 with artificial data. Figure 5a shows the output of the pre-processing algorithm for an input signal of 1 Hz. There are two interesting features of this image. Firstly the amplitude of the processed signal is significantly attenuated with respect to the input, and secondly the majority of the processed signal is in the negative half of the plane. Figure 5b shows the output of the pre-processing algorithm with an input signal with identical amplitudes oscillating at 5 Hz. Due to the increase in frequency of the input signal, the pre-processing algorithm measures sharper slopes and the output of the algorithm produces larger and mostly positive outputs.

Taking the product of the slopes of the left and right halves of the sliding window, therefore, introduces relevant context that assists with the classification problem. Figure 4f demonstrates that in some cases a QRS complex can assume the form of a negative peak. A classifier provided with only the left and right slopes as two features may, therefore, need to be capable of solving the XOR problem, and strictly linear classifiers are incapable of doing this. Researchers typically take the absolute value of filtered inputs to solve this problem, but this removes the trending directional context of the signal.

2) Artefact: Figure 6 shows the output of the pre-processing algorithm under the influence of artefact. In this figure, the regions highlighted in grey are the times at which QRS points have been annotated.

Baseline wander can be observed in Figure 6a. The algorithm copes with these artefacts for the same reasons that the algorithm copes with the slow wave in Figure 4d. However, the cause of these wave shapes is different in these two cases — the slow wave in Figure 4d is due to the unhealthy ECG, whereas the contribution in Figure 6a is due to artefact.

Figure 6b shows the the ECG under the influence of high frequency artefact. The amplitude of the artefact in the ECG is comparable to the amplitude of the QRS complex, but the pre-processing algorithm has eliminated a significant proportion of

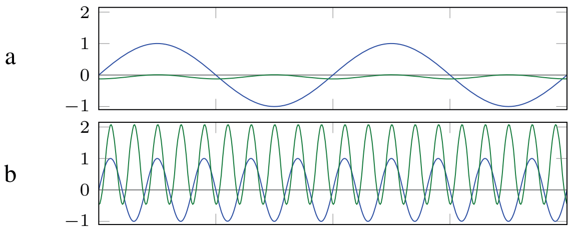

Figure 5: Pre-processing algorithm on artificial input data.

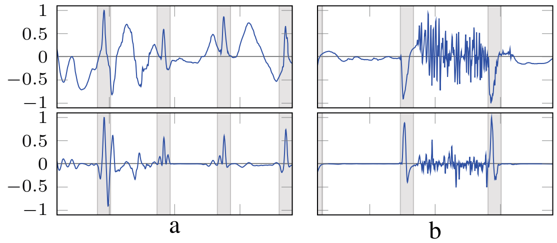

Figure 6: Example of artefact in the ECG.

this in its processed representation.

Typically, researchers cope with these kinds or artefact by employing multiple stages of high-order digital filters (low-pass, high-pass, pre-emphasis). We have not employed such digital filters in our pre-processing steps. Even so, our results are good, and we expect that introducing such noise removing techniques would improve our results even further.

C. Post-processing

There are a number of differences between the post-processing routine used here and others used in the QRS detection literature. Firstly, we do not employ 'back tracking' to select QRS points when none have been detected for a certain length of time. This must often be performed to increase the sensitivity of other algorithms to detections to permissible levels. We also select the period of which has contributed to the highest cumulative probability of being a QRS point within a wider remit of plausible candidates. This has a favourable 'moving average' effect that may reject spurious periods of high probability, and, as classifiers are trained on examples of QRS points, this is a very reasonable method with which to make decisions. Finally, we do not adapt our threshold during decision making.

Our post-processing algorithm is considerably simpler than those employed by other researchers (e.g. [9]). In these publications, the researchers must perform multiple stages of post-processing each of which introduces new metrics and adaptive thresholds in order to produce reliable predictions. Our approach, however, copes with this context well and even with its relative simplicity performs competitively or better than these more elaborate methods.

D. Analysis of errors

In previous sections, we have demonstrated the tolerance of our framework towards artefacts. In some circumstances the intensity of the artefacts are such that our pre-processing algorithm is overwhelmed and false positives are classified; this is a factor common to all detection algorithms. We now discuss the more interesting circumstances that can lead to false negative classifications of the QRS complex.

Figure 7 shows two examples of QRS points that were not detected by our algorithm. The top row shows the raw ECG signal trace and the bottom row shows the probabilities of QRS complexes from by the LR (blue) and SVM (green) classifiers. The regions shaded in red are the times annotated as QRS complexes but which our algorithm has failed to detect.

In all these false negative cases one very striking feature can be observed — these missed QRS points demonstrate very few characteristics of the healthy ECG shown in Figure 1. In Figures 7a the QRS complex is seemingly absent from the highlighted region, and both classifiers yield only low probabilities of positive detections.

In Figure 7b, we show a false negative that somewhat resembles a QRS complex. This was not identified by the algorithm due to its 'slow' nature. Comparing its width to the width of the preceding QRS complex shows it to be abnormal, and our pre-processing algorithm attenuated the intensity of its representation accordingly.

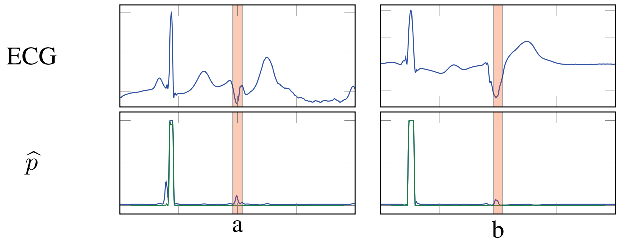

Figure 7: Demonstration of false negatives.

In this work, however, we have only considered single-lead ECG which does not always yield the most representative view of heart activity, and it is likely that by considering a greater number of ECG channels fewer errors would be obtained.

IV. Conclusion

We have presented a QRS detection algorithm which accurately detects the QRS complex from the ECG. The performance of the detection algorithm is competitive, and sometimes superior, to the performance of the state of the art in QRS detection, but one of the advantages of our procedure is that our results are obtained after applying much simpler pre- and post-processing techniques. Our framework also yields a significantly lower difference between the sensitivity and precision of detection than are typically cited in similar work.

Acknowledgement

This work was performed under the SPHERE IRC funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1.

References

[1] N. Twomey *et al.*, "Automated detection of perturbed cardiac physiology during oral food allergen challenge in children," *IEEE Journal of Biomedical and Health Informatics*, 2013.

[2] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, 2000.

[3] W. F. Ganong *et al.*, *Review of medical physiology*. McGraw-Hill Medical New York:, 2005, vol. 21.

[4] V. Vapnik, *The nature of statistical learning theory*. springer, 2000.

[5] B. Scholkopf *et al.*, *Learning with kernels*. MIT press Cambridge, 2002.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, 2011.

[7] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, 1999.

[8] J. Bergstra *et al.*, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, 2012.

[9] V. X. Afonso *et al.*, "ECG beat detection using filter banks," *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 2, 1999.

[10] V. Vapnik *et al.*, *Estimation of dependences based on empirical data*. Springer, 2006.

[11] Z. Zidelmal *et al.*, "Qrs detection based on wavelet coefficients," *Computer methods and programs in biomedicine*, vol. 107, no. 3.

[12] C. Zeng *et al.*, "Qrs complex detection using combination of mexican-hat wavelet and complex morlet wavelet." *Journal of Computers*, vol. 8, no. 11, 2013.