Chapter 3

Signal processing and feature selection preprocessing for classification in noisy healthcare data

Qiao Li, Chengyu Liu, Julien Oster and Gari D. Clifford

3.1 Introduction

AQ1

Although current healthcare practice is centered on human expert assessment of the correlations between parameter values and symptoms, there is a growing awareness within medical communities that the enormous quantity and variety of data available cannot be effectively assimilated and processed without automated or semi-automated assistance [1]. Automated systems have been in place in the intensive care unit (ICU), the operating room (OR) and clinical ward for several decades, including automated arrhythmia analysis of the bedside electrocardiogram (ECG) and low (or high) oxygen saturation warnings from the photoplethysmograph (PPG). However, each device acts in an isolated fashion with no reference to related signals or an individual's prior medical information, such as genetics or medical history. Since modern physiological monitoring devices are tuned to be highly sensitive, but prone to noise, a paradigm shift in monitoring technology is required, which allows for more intelligence in the device and less expert oversight [2]. Artifacts, noise and missing values are the main reasons of the high levels of false alarms [3]. Meanwhile, the explosion of mHealth in both abundant and resource-constrained countries is both a cause for concern and celebration [4–7]. While mHealth clearly has the potential to deliver information and diagnostic decision support to the poorly trained, it is not appropriate to simply translate the technologies which the trained clinician uses into the hands of nonexperts. In particular, it is important that the explosion of access does not lead to a flooding of the medical system with low-quality data and false negatives. Although telehealth has the potential to connect remote users with little training to trained experts, with the patient-to-doctor ratio being as low as 50,000:1 in parts of low-income countries, automated algorithms will be essential to cope with the number of recordings that are likely to be made available. Moreover, since the greatest (and often the only) chance for improving the quality of physiological data is at source, a rapid feedback to the recordist or user concerning the clinical viability of the data is needed. Therefore, data screening must occur at the front end using automated algorithms, prompting the user to retake recordings when quality is low.

In order to provide information for medical experts (or automated decision support systems) to make choices concerning patient care, the wealth of available data must be reduced to a set of distinct concepts and features. Although many parameters are derived from patient data "on the fly" and recorded for later review, trust metrics or signal quality measures associated with these parameters are rarely stored. Therefore, it is difficult to ascertain the credibility of a given parameter unless the original data from which the parameter was derived are available, either to visually verify the data or in order to derive independent quality metrics.

Noise reduction algorithms often introduce misleading distortions in medical time-series data and, therefore, they should be applied only when the data are determined to be too noisy for a feature extraction algorithm to be applied accurately. However, it is often necessary to extract features and compare them with a population norm, or a patient's history, in order to determine whether significant amounts of noise are present. A method for simultaneously (or recursively) extracting features and estimating noise levels is, therefore, appropriate.

Since robust methods for dealing with noisy data are not always available, it is sometimes more appropriate to define a signal quality measure for a given data stream, and simply ignore the segments of data that have a signal quality below a given value. However, metrics for signal quality are both signal and application specific. Signal quality indices (SQIs) can generally be constructed by thresholding on known physiological limits such as the maximum field strength for the ECG, the maximum rate of change of the blood pressure or the distribution of energy in the frequency domain. However, it is the relationship between physiological parameters that provides the greatest opportunity to construct SQIs. For example, if heartbeats are detected in several ECGs and/or pulsatile waveforms within an expected period of time, all signals can be considered to be of reasonable quality. In Li *et al.* [8], we calibrated a set of ECG signal quality metrics (based on statistical, temporal, spectral and cross-spectral features of the ECG), so that a given value of an SQI metric was equated to known error in heart rate. A similar approach was also taken AQ2 to ABP, and hence error bounds in derived estimates that rely on heart rate and blood pressure (such as the cardiac output) can easily be estimated from the standard compound error formula. Generally, data in the ICU are processed in isolation from other parameters and signal quality labels are therefore rarely constructed with reference to other signals. In our approach to SQI derivation, we have concentrated on the relationships between signals, such as the transit time between the ECG and the ABP [9] and the inter-ECG lead relationships [8]. By comparing related signals and thresholding these relationships on known physiological limits, it is possible to determine whether the data are logically consistent. Since it is rare that a sequence of extracted features will randomly manifest in a physiologically plausible manner, internal consistency between signals can indicate high signal quality on the contributing leads.

Throughout this chapter, we illustrate our approach to signal processing and feature selection preprocessing for atrial fibrillation (AF) detection in noisy environment. AF is the most common cardiac arrhythmia, whose prevalence is 0.4–1% in the general population and increases with age [10]. AF is associated with a fivefold increased risk of stroke, and one in six strokes occurs in patients with AF. This pathology can be symptomatic, (e.g., palpitation and fatigue) but can also be asymptomatic, which makes AF currently under-diagnosed. ECG signals acquired during ambulatory recordings and more specifically with mHealth applications are prone to noise and artifacts. Such recordings are also performed in an uncontrolled environment and by nonexperts.

The goal of this study is therefore to assess the preprocessing algorithms and the influence of noise on the estimation of RR intervals and how these noisy estimates of the RR time-series will impact the detection of AF episodes by state-of-the-art automated algorithms.

3.2 Preprocessing and database

3.2.1 QRS detection

Three popular QRS detectors were used to detect the QRS complex of ECG.

1. jqrs: [11,12] consists of a window-based peak energy detector. The original band-pass filter has been replaced with a QRS matched filter (Mexican hat) and an additional heuristic ensuring no detection was made during flat lines. A search-back procedure is also allowed in case of suspected missed beats.
2. gqrs: (available on Physionet; https://www.physionet.org/physiotools/wag/gqrs-1.htm), which consists of a QRS matched filter with a custom built set of heuristics (such as search back). It has been designed by George Moody, and is freely available on Physionet, but does not have an associated publication.
3. wqrs: [13] consists of a low-pass filter, a nonlinearly scaled curve length transformation and decision rules. It is freely available on Physionet.

A majority voting of the results of the three detectors was evaluated to calculate the beat-by-beat RR intervals.

3.2.2 Signal quality assessment

SQI of ECG was implemented based on a machine learning approach, which combines several simple quality metrics [14,15]. Of these, bSQI is the most important one and it consists of the comparison of two different peak detectors, jqrs and wqrs, one (wqrs) being more sensitive to noise than the other (jqrs). bSQI therefore indicates when the R peak detection is precise, and was used in this study. bSQI was computed on a 10-s window and sliding the window forward every second to get the second-by-second bSQI.

3.2.3 Datasets

Two databases were used in this study, the MIT-BIH atrial fibrillation database (AFDB) and the long-term AF database (LTAFDB), which are open access at www.physionet.org

The AFDB includes 25 ECG recordings of human subjects with AF (mostly paroxysmal). Of these, 23 records include two ECG signals with rhythm and unaudited beat annotations. The rest two records are represented only by the rhythm and annotation files without ECG signals and are eliminated from this study. The individual ECG recordings are each 10 hours in duration, and contain two ECG signals each sampled at 250 samples per second with 12-bit resolution over a range of ±10 millivolts. The rhythm annotation files were prepared manually; these contain rhythm annotations of types AFIB (atrial fibrillation), AFL (atrial flutter), J (AV junctional rhythm) and N (used to indicate all other rhythms). The LTAFDB includes 84 long-term ECG recordings of subjects with paroxysmal or sustained AF. Each record contains two simultaneously recorded ECG signals digitized at 128 Hz with 12-bit resolution over a 20 mV range; record durations vary but are typically 24–25 hours. The types of rhythm annotations include AFIB (atrial fibrillation), N (normal sinus rhythm), SVTA (supraventricular tachyarrhythmia), VT (ventricular tachycardia), B (ventricular bigeminy), T (ventricular trigeminy), IVR (idioventricular rhythm), AB (atrial bigeminy) and SBR (sinus bradycardia). In this study, we regard the AFIB annotation as AF (1) and all other rhythm annotations as Non-AF (0).

The design of machine learning algorithm of AF detection included a development phase and a validation phase. The AFDB was used in the development phase and the LTAFDB was used in the validation phase. Here we recommend to validate the robustness of the algorithm on an unseen database which is different from the development phase.

In the development phase, the ECG in AFDB was analyzed by the three QRS detectors and a majority voting was performed to calculate beat-by-beat RR intervals. The first channel of ECG was analyzed except record 07162 which the voltage of QRS complex is low in the first channel and the second channel was used. The AF and Non-AF rhythms were marked segment-by-segment by a 30-s length analysis window. AQ4 Here we selected a 30-s window due to that the AF events usually last 30 s or even longer to be considered from the clinical point of view [16]. Rhythms with lengths shorter than 30 s were discarded. The bSQI was computed on a second-by-second basis, and a unique score was derived for each window by the median of the bSQI over the window. In order to avoid the influence of noise during the development phase, the low quality segments with a median of bSQI lower than 0.85 were removed from the dataset. A resultant dataset with total 26,925 high quality segments was extracted from AFDB, including 10,541 AF segments and 16,384 Non-AF segments. The dataset was then split randomly into training set and test set, stratified by patients rather than by segments, as shown in Table 3.1. Stratification by patients ensures that the training set and test set contain mutually exclusive patients and reduces the chances of over-training. A K-fold cross validation, also stratified by patients, was also performed to avoid overfitting during the development phase.

In the validation phase, the first channel of ECG in LTAFDB was analyzed by three QRS detectors except records 00, 24, 56 and 62, in which the first channel was very noisy and so the second channel was used. Note we did not eliminate the noisy segments in the validation phase, so that the validation statistics reflect both a real-world scenario, with previously unseen patients containing noisy data. Importantly,

Table 3.1 Datasets using in this study

	Development phase (AFDB)						**Validation phase (LTAFDB)**	
	Training set (12 cases)		**Test set (11 cases)**		**Total (23 cases)**		**Total (84 cases)**	
	AF	**Non-AF**	**AF**	**Non-AF**	**AF**	**Non-AF**	**AF**	**Non-AF**
Segments	5,327	8,639	5,214	7,745	10,541	16,384	118,473	103,498
Total	13,966		12,959		26,925		221,971	

an entirely separate database was used, ensuring differences in patient population and recording techniques. A validation dataset with total 221,971 segments was extracted from LTAFDB, including 118,473 AF segments and 103,498 Non-AF segments.

3.2.4 Adding realistic noise to known data

To evaluate the influence of the noise to AF detection, we added the muscle artifact (MA) noise, simulated using the fecgsyn toolbox [17], to each of the ECG signals in the LTAFDB in the validation phase. Simulations with different SNR levels (24, 21, 18, 15, 12, 9, 6, 3, 0, −3 dB) were performed.

AQ5

3.3 Feature extraction

Feature extraction is the process of reducing a set of raw or preprocessed data into a smaller set of quantities (features) that represent the key qualities of the data. Features should be chosen (or found) such that they possess highly different values for each class of data that requires identification (or classification). Since there is an almost infinite number of statistics and metrics that can be extracted from a given set of data, prior knowledge of the system (e.g., physiology or noise profiles under certain conditions) is often used to guide feature extraction. For example, AF is characterized by a chaotic electrical conduction through the AV node and ventricular response, resulting in an unpredictable depolarization of the ventricles, and therefore the RR interval time-series. It is not completely unpredictable, and a probabilistic modeling of the RR intervals during AF episodes has been recently suggested [18]. The use of the statistics of RR intervals for the detection of AF episodes has been proven to be possible, and several methods have been proposed [19–21]. In this study, we have chosen to use a superset of the 14 RR interval time-series features proposed in these earlier studies. Although this may not be exhaustive, it provides a tractable list from which we can then perform feature selection (to remove redundant or suboptimal features).

3.3.1 Time-domain features

The mean value (mRR), minimum value (minRR) and maximum value (maxRR) of RR intervals of the current RR segment, the median value of HR (medHR), the standard deviation of RR intervals (SDNN), the percentage of RR intervals larger than 50 ms (PNN50) and the square root of the mean squared differences of successive RR intervals (RMSSD) were used as time-domain features [22].

3.3.2 Frequency-domain features

Burg's autoregressive approach (with an order of 6) was used to produce the power spectrum for the RR segment. The power spectrum was integrated over two frequency ranges: the low-frequency power (0.04–0.15 Hz) and the high-frequency power (0.15 to 0.40 Hz). The normalized low-frequency power (LF_n), normalized high-frequency power (HF_n) and the ratio of low-frequency power to high-frequency power (LF/HF) were used as the frequency-domain features [22].

3.3.3 Nonlinear features

Coefficient of sample entropy (COSEn) and normalized fuzzy entropy (NFEn) were used as nonlinear features [23–25], with an embedding dimension $m = 1$. For a detailed discussion process for COSEn and NFEn please refer to the Appendix 1.

AQ6 MAD feature [26], defined as the median of the variation in the absolute standard deviation from mean of heart rate in three adjacent RR segments with same length, was used as another nonlinear feature. An AF evidence feature (AFEv), as a numeric representation of the Lorenz plot, a two-dimensional histogram, was also used [27, 28]. The MAD method requires that the length of RR segment should be perfectly divisible by 3. Therefore, each window was truncated so that the number of RR intervals was rounded to be as large as possible while being exactly divisible by three.

3.4 Feature selection

Feature selection is primarily performed to select relevant and informative features. It can have other motivations, including [29]:

- general data reduction, to limit storage requirements and increase algorithm speed;
- feature set reduction, to save resources in the next round of data collection or during utilization;
- performance improvement, to gain in predictive accuracy;
- data understanding, to gain knowledge about the process that generated the data or simply visualize the data.

There are three main categories of feature selection algorithms: filters, wrappers and embedded methods. Filter methods, or called feature ranking methods, provide a complete order of the features using a relevance index, including correlation coefficients, classical test statistics (t-test, F-test, chi-squared, etc.), mutual information and information theory. Wrappers and embedded methods involve the predictor as part of the selection process. Wrappers utilize a learning machine as a "black box" to score subsets of features according to their predictive power. Embedded methods perform feature selection in the process of training and are usually specific to given learning machines.

In this study, we tested two feature selection methods corresponding to two machine learning algorithms, logistic regression and support vector machine.

3.4.1 Forward likelihood ratio selection for logistic regression

Logistic regression (LR) is a statistical model for classification, which identifies the impact of multiple independent variables in predicting the membership of one of the multiple dependent categories. For binary logistic regression (BLR), the number of the dependent categories was limited as two. BLR can be considered an extension of linear regression, which struggles with dichotomous problems. This difficulty is overcome by applying a mathematical transformation of the output of the classifier, transforming it into a bounded value between 0 and 1 more appropriate for binary predictions.

In the current study, the output variable Y is a positive (1) or negative (0) diagnosis for AF: the posterior probability $P\ (y|X)$ for the input feature vector X is modelled by a logistic function, as follows:

$$P(Y=0|X)=\frac{1}{1+\exp(w^TX)} \tag{3.1}$$

$$P(Y=1|X)=\frac{\exp(w^TX)}{1+\exp(w^TX)} \tag{3.2}$$

where w is the vector of the regression coefficients.

The sigmoid function $S(t)$ is usually used as the standard logistic function and is defined as:

$$S(t)=\frac{1}{1+\exp(-t)} \tag{3.3}$$

Likelihood ratio (also named as odds ratio) is defined as the natural logarithm of (3.1) and (3.2). Thus a linear dependence between conditional probabilities and predictive variables is established as:

$$\ln\frac{P(Y=1|X)}{P(Y=0|X)}=\ln\frac{\exp(w^TX)\big/(1+\exp(w^TX))}{1\big/(1+\exp(w^TX))}=w^TX \tag{3.4}$$

From (3.4), if $P(Y=1|X)=P(Y=0|X)$, i.e., the probabilities of predicting AF and Non-AF equal, the output of w^TX will be 0. So we can use the training set to train the BLR model, determining the selected feature vector X and their regression coefficients vector w. Then we can set $z=w^TX$ and calculate the outputs for the RR segments of test set, predicating them as AF segments if $z>0$ and as Non-AF segments if else.

The aforementioned BLR analysis was performed on SPSS version 19 to explore the potential predictable features for AF detection. All 14 features were tested. Forward likelihood ratio selection was used. Initially there are no features in the model. Then the feature with the largest likelihood was selected into the model. If the statistical difference was significant with the adding of this feature, the feature was reserved as a predictable feature. Then the feature with the largest likelihood in the remaining features was selected into the model and the comparison was also performed. The selection will be ended if the newly added feature could not significantly improve the AF prediction results. The limit with this method is that it can be too greedy: features are fully added at each step, so correlated predictors are unlikely to be included in the model.

3.4.2 Recursive feature elimination for support vector machine

The fundamental idea of support vector machine (SVM) classifier is the construction of the optimal hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$, which separates different classes with maximal margin [29,30].

The maximal margin can be defined as maximization of the minimum distance between vectors and the hyperplane:

$$\max_{\mathbf{w},b} \min \left\{ \|\mathbf{x} - \mathbf{x}_i\| : \mathbf{w}^T\mathbf{x} + b = 0, i = 1, \ldots, m \right\} \tag{3.5}$$

The $\mathbf{w}$ and b can be rescaled in a way that the point closest to the hyperplane lies on a hyperplane $\mathbf{w}^T\mathbf{x} + b = \pm 1$. Hence for every x_i we get: $\mathrm{y}_i[\mathbf{w}^T\mathbf{x}_i + b] \geq 1$, so the width of the margin is equal to $2/||\mathbf{w}||$. Equation (3.5) then can be restated as the optimization problem of objective function:

$$\min_{\mathbf{w},b} \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.6}$$

With the following constraints:

$$\mathrm{y}_i[\mathbf{w}^T\mathbf{x}_i + b] \geq 1, i = 1, \ldots, m \tag{3.7}$$

To solve it, a Lagrangian is constructed:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i(y_i[\mathbf{x}_i^T\mathbf{w} + b] - 1) \tag{3.8}$$

where $\alpha_i > 0$ are Lagrange multipliers. Its minimization leads to:

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{3.9}$$

According to the Karush-Kuhn-Thucker conditions [31],

$$\alpha_i(\mathrm{y}_i[\mathbf{x}_i^T\mathbf{w} + b] - 1) = 0, i = 1, \ldots, m \tag{3.10}$$

The non-zero α_i corresponds to $\mathrm{y}_i[\mathbf{x}_i^T\mathbf{w} + b] = 1$. It means that the vectors which lie on the margin play the crucial role in the solution of the optimization problem. Such vectors are called support vectors.

After some substitutions the optimization problem can be transformed to the dual optimization problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{3.11}$$

with constraints:

$$\alpha_i > 0 \quad i = 1, \ldots, m, \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.12}$$

Using the solution of this problem the decision function can be written as:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}^T \mathbf{x}_i + b\right) \tag{3.13}$$

To replace the dot product $\mathbf{x}^T\mathbf{x}'$ by a kernel function $k(\mathbf{x}, \mathbf{x}')$, it extends the linear SVM to a nonlinear SVM. The new decision function is:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{3.14}$$

In this study, we used the Gaussian kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp[-\gamma \left\|\mathbf{x} - \mathbf{x}'\right\|^2] \tag{3.15}$$

A standardization of the features is necessary before SVM training. The following centering and scaling of the data is used: $x_i' = (x_i - \mu_i)/\sigma_i$, where μ_i and σ_i are the mean and the standard deviation of feature x_i over training set. Note that the same μ_i and σ_i over the training set are used over the test set too.

A recursive feature elimination (RFE) algorithm was used for feature selection which was proposed by Guyon *et al.* [32]. The RFE algorithm method attempts to find the best subset of size σ $(\sigma < N)$ by a kind of greedy backward selection. It operates by trying to choose the σ features which lead to the largest margin of class separation by a SVM classifier. This combinatorial problem is solved in a greedy fashion at each iteration of training by removing the input dimension that decreases the margin the least until only σ input dimensions remain.

For a nonlinear SVM, the margin is inversely proportional to the value $W^2(\alpha) := \sum \alpha_k\alpha_l y_k y_l k(\mathbf{x}_k, \mathbf{x}_l)$. The algorithm thus tries to remove features that lead to small values of this variable. An iterative procedure was performed as below.

```
Repeat
    Train a SVM on training set
    Given the solution $\alpha$, calculate $W^2_{(-p)}(\alpha)$ for each feature $p$:
        $W^2_{(-p)}(\alpha) = \sum \alpha_k\alpha_l y_k y_l k(\mathbf{x}_k^{-p}, \mathbf{x}_l^{-p})$
        (where $\mathbf{x}_k^{-p}$ means training point $k$ with feature $p$ removed)
    Remove the feature with smallest value of $W^2(\alpha) - W^2_{-p}(\alpha)$
Until $\sigma$ feature remains.
```

3.5 Evaluation metrics

The accuracy of AF predictor can be evaluated by the following indices:

- Sensitivity: $Se = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$
- Specificity: $Sp = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP})$
- Accuracy: $Acc = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN})$
- AUROC: the area under the receiver operating characteristic (ROC) Curve

where TP is the number of true positive, TN is the number of true negative, FP is the number of false positive and FN is the number of false negative.

3.6 Results

3.6.1 Feature results comparison between AF and Non-AF

Table 3.2 shows the average values of all features for the AF and Non-AF RR segments (with one standard deviation). A group *t*-test demonstrates that are significant differences of all features ($P < 0.01$) between the two groups except for the LF/HF index.

Table 3.2 Statistical group t-test results for comparison between the AF and Non-AF groups

Variable	**AF**	**Non-AF**
Number of RR segments	10,541	16,384
mRR (s)	0.68 ± 0.14*	0.83 ± 0.16
minRR (s)	0.44 ± 0.10*	0.70 ± 0.22
maxRR (s)	1.07± 0.33*	0.96 ± 0.34
medHR (beats/min)	96 ± 21*	75 ± 16
SDNN (s)	0.15 ± 0.06*	0.05 ± 0.08
PNN50 (%)	73 ± 13*	14 ± 21
RMSSD (s)	0.20 ± 0.09*	0.08 ± 0.12
$\mathrm{LF_n}$	0.38 ± 0.14*	0.29 ± 0.21
$\mathrm{HF_n}$	0.62 ± 0.14*	0.71 ± 0.21
LF/HF	0.70 ± 0.46	0.70 ± 1.26
COSEn	−0.93 ± 0.52*	−2.10 ± 0.87
NFEn	0.55 ± 1.00*	−3.03 ± 1.65
MAD ($\times 10^{-5}$)	21.2 ± 9.7*	2.5 ± 8.3
AFEv	32.5 ± 9.1*	−14.8 ± 14.3

Note: Data are presented by mean ± standard deviation (SD). "*" means significant differences compared with Non-AF group.

Table 3.3 Feature selection results for the training set and the corresponding regression coefficients of binary logistic regression using forward likelihood ratio method. The results are given for each regression step

Regression step	**Regression coefficients for the selected variables**								
	Constant	**PNN50**	**AFEv**	**MAD**	**NFEn**	**COSEn**	**mRR**	**HFn**	**LF/HF**
1	−5.137	0.108	0	0	0	0	0	0	0
2	−5.184	0.030	0.235	0	0	0	0	0	0
3	−5.231	0.058	0.230	−9,481	0	0	0	0	0
4	−3.659	0.046	0.202	−10,365	0.797	0	0	0	0
5	−6.711	0.044	0.150	−12,518	3.210	−4.495	0	0	0
6	−10.193	0.019	0.180	−10,730	3.656	−5.419	4.187	0	0
7	−9.903	0.020	0.174	−10,617	3.764	−5.669	4.754	−1.396	0
8	−6.396	0.018	0.174	−10,231	3.688	−5.545	5.194	−5.632	−1.282

3.6.2 Model development phase

3.6.2.1 Logistic regression result

Table 3.3 shows the feature selection results for the training set and the corresponding regression coefficients of BLR using the forward likelihood ratio method. The results are given for each regression step. After eight regression steps, eight features were identified as the predictable features, including PNN50, AFEv, MAD, NFEn, COSEn, mRR, HFn and LF/HF in turn. As shown in Table 3.3, the final prediction formula for AF segment is:

$$\begin{aligned} z = w^T X \; = \; & -6.396 + 0.018 \times \text{PNN50} + 0.174 \times \text{AFEv} - 10231 \\ & \times \text{MAD} + 3.688 \times \text{NFEn} - 5.545 \times \text{COSEn} + 5.194 \times \text{mRR} \\ & - 5.632 \times \text{HFn} - 1.282 \times \text{LF/HF} \end{aligned} \tag{3.16}$$

Table 3.4 shows the results of TP, FN, FP and TN numbers and the three indices (Se, Sp and Acc) for both training and test sets with the evaluation for each regression step. Using (3.16), the final AF prediction results were 99.4% for Se, 98.8% for Sp and 99.0% for Acc for the training set, and were 97.1% for Se, 94.9% for Sp and 95.8% for Acc for the test set. AQ7

K-fold cross-validation

Table 3.5 shows the results for K-fold cross validation (K = 9). For each of the nine subsets, the selected features and the corresponding regression coefficients of the BLR model using the forward likelihood ratio method are given, as well as the evaluation results for both training and test sets. Finally, the results for voting together the nine BLR models are given, with a final *Se* of 98.5%, *Sp* of 97.9% and *Acc* of 98.1% for all 26,925 RR segments.

Table 3.4 Results of the TP, FN, FP and TN numbers and the three indices (Se, Sp and Acc) for both training and test sets with the evaluation for each regression step

Regression step	**Training data**							**Test data**						
	TP	**FN**	**FP**	**TN**	***Se* (%)**	***Sp* (%)**	***Acc* (%)**	**TP**	**FN**	**FP**	**TN**	***Se* (%)**	***Sp* (%)**	***Acc* (%)**
1	5,136	191	356	8,283	96.4	95.9	96.1	4,990	224	1,171	6,574	95.7	84.9	89.2
2	5,299	28	157	8,482	99.5	98.2	98.7	5,071	143	384	7,361	97.3	95.0	95.9
3	5,288	39	126	8,513	99.3	98.5	98.8	5,068	146	424	7,321	97.2	94.5	95.6
4	5,292	35	128	8,511	99.3	98.5	98.8	5,069	145	387	7,358	97.2	95.0	95.9
5	5,299	28	116	8,523	99.5	98.7	99.0	5,067	147	421	7,324	97.2	94.6	95.6
6	5,297	30	101	8,538	99.4	98.8	99.1	5,042	172	406	7,339	96.7	94.8	95.5
7	5,298	29	105	8,534	99.5	98.8	99.0	5,054	160	395	7,350	96.9	94.9	95.7
8	5,294	33	100	8,539	99.4	98.8	99.0	5,063	151	397	7,348	97.1	94.9	95.8

Table 3.5 The results for the K-fold cross-validation

Variable	**Subsets for K-fold cross-validation**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Training									
TP	7,583	10,089	9,290	9,210	9,555	9,617	9,056	9,184	9,399
FN	143	169	131	143	132	174	170	134	149
FP	273	314	276	318	261	325	286	283	334
TN	15,401	12,894	14,835	14,894	13,388	13,301	14,981	14,884	13,824
Se (%)	98.1	98.4	98.6	98.5	98.6	98.2	98.2	98.6	98.4
Sp (%)	98.3	97.6	98.2	97.9	98.1	97.6	98.1	98.1	97.6
Acc (%)	98.2	97.9	98.3	98.1	98.3	97.9	98.1	98.3	98.0
Test									
TP	2,680	272	1,009	1,175	854	750	1,237	1,214	990
FN	135	11	111	13	0	0	78	9	3
FP	12	12	24	51	155	25	8	128	24
TN	698	3,164	1,249	1,121	2,580	2,733	1,109	1,089	2,202
Se (%)	95.2	96.1	90.1	98.9	100.0	100.0	94.1	99.3	99.7
Sp (%)	98.3	99.6	98.1	95.6	94.3	99.1	99.3	89.5	98.9
Acc (%)	95.8	99.3	94.4	97.3	95.7	99.3	96.5	94.4	99.2
Summary of all K models									
Total TP					10,181				
Total FN					360				
Total FP					439				
Total TN					15,945				
Mean *Se* (%)					97.0 ± 3.4				
Mean *Sp* (%)					97.0 ± 3.3				
Mean *Acc* (%)					96.9 ± 2.0				
Voting all K models									
TP					10,389				
FN					152				
FP					358				
TN					16,026				
Se (%)					98.6				
Sp (%)					97.8				
Acc (%)					98.1				

3.6.2.2 SVM result

RFE feature selection

The result of RFE feature selection for SVM algorithm is shown in Table 3.6. In the beginning all features are in the model. The order of feature removing is LFn, HFn, LF/HF, MAD, COSEn, mRR, medHR, NFEn, RMSSD, PNN50, maxRR, SDNN and minRR during the iteration. AFEv is the last one keeping in the model. After the sixth iteration, the AUROC gets the maximum on the test set. There are eight features left in the model, AFEv, minRR, SDNN, maxRR, PNN50, RMSSD, NFEn and medHR.

Table 3.6 Result of RFE feature selection

AQ8

Iterate step	Removed feature each step	Training set				Test set			
		Se	Sp	Acc	AUROC	Se	Sp	Acc	AUROC
0	–	99.51	99.24	99.34	99.85	96.36	96.75	96.59	99.30
1	LFn	99.53	99.24	99.35	99.85	96.36	96.73	96.58	99.27
2	HFn	99.47	99.25	99.33	99.86	96.43	96.69	96.59	99.22
3	LF/HF	99.49	99.25	99.34	99.85	96.16	96.79	96.54	99.20
4	MAD	99.42	99.25	99.31	99.85	96.13	97.20	96.77	99.26
5	COSEn	99.42	99.18	99.27	99.86	96.28	97.24	96.85	99.29
6	mRR	99.38	99.18	99.26	99.85	96.36	97.17	96.84	**99.31**
7	medHR	99.40	99.20	99.28	99.81	96.24	96.53	96.41	99.11
8	NFEn	99.38	99.11	99.21	99.77	96.38	96.42	96.40	99.04
9	RMSSD	99.34	99.10	99.19	99.74	96.14	96.40	96.30	98.99
10	PNN50	99.32	99.05	99.16	99.72	96.99	96.41	96.64	99.16
11	maxRR	99.31	98.88	99.04	99.72	97.30	96.23	96.66	99.04
12	SDNN	99.27	98.72	98.93	99.62	97.62	95.20	96.17	98.47
13	minRR	99.32	98.33	98.71	99.31	98.12	94.40	95.89	98.05

Table 3.7 Result of K-fold cross-validation

K-fold iterate	Training set (8 folds)				Test set (1 fold)			
	Se	Sp	Acc	AUROC	Se	Sp	Acc	AUROC
1	98.46	98.87	98.74	99.79	93.57	98.87	94.64	99.04
2	98.71	98.45	98.56	99.75	94.70	99.28	98.90	99.45
3	98.93	99.05	99.00	99.75	85.27	98.51	92.31	98.37
4	98.59	98.53	98.55	99.75	98.99	97.10	98.05	99.78
5	98.70	98.83	98.77	99.76	99.88	96.67	97.44	99.74
6	98.53	98.38	98.44	99.70	99.73	99.93	99.89	100.00
7	98.53	98.76	98.67	99.74	93.38	99.28	96.09	99.68
8	98.84	98.83	98.84	99.80	99.35	90.06	94.71	99.56
9	98.60	98.50	98.54	99.76	99.70	98.92	99.16	99.77
Mean	98.65	98.69	98.68	99.76	96.06	97.62	96.80	99.49
Std	0.16	0.23	0.18	0.03	4.90	3.03	2.53	0.50

K-fold cross validation

The result of K-fold cross validation is shown in Table 3.7. Note that we used ninefold rather than 10 fold here is due to that the odd number is convenient for majority voting. After we got the nine SVM models, we classified the whole dataset again using the nine models and compared the result between the mean and the majority voting of the nine models. The result is shown in Table 3.8. It shows that the Acc of majority voting is slightly better than that of mean (98.66% vs 98.50%).

Table 3.8 Comparison of mean and majority voting of nine models on the whole dataset

	Se	**Sp**	**Acc**
Mean of K models	98.31 ± 0.64	98.62 ± 0.25	98.50 ± 0.14
Voting of K models	98.65	98.66	98.66

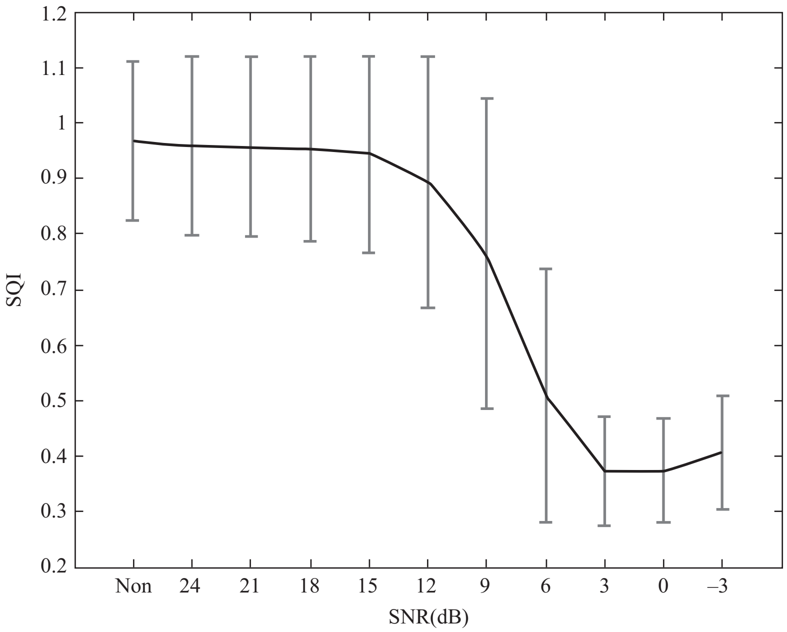

Figure 3.1 SQI of the dataset along with the various SNR of adding noise

3.6.3 Model validation phase

The models which were established in the development phase were validated on the unseen LTAFDB dataset with various SNR of adding noise and by different QRS detectors.

Figure 3.1 shows the mean and standard deviation of SQI of the dataset along with the various SNR of adding noise. When the SNR ranges from 24 dB to 15 dB, the SQI keeps at a high level (above 0.9) with a slight dropping, since these levels of adding noise have little influence to the QRS detection. Accompanied with the drop of SNR from 12 dB to 3 dB, the SQI drops obviously from 0.89 to 0.37. However, when the SNR keeps dropping to 0 dB and −3 dB, the SQI keeps at a low level (below 0.4)

Table 3.9 The result of Logistic regression models on LTAFDB dataset with and without adding noise (mean of K models)

Adding noise (dB)	jqrs			gqrs			wqrs			Voting (QRS)		
	Se	Sp	Acc	Se	Sp	Acc	Se	Sp	Acc	Se	Sp	Acc
Non	98.07	93.46	95.94	97.82	91.41	94.83	98.08	90.80	94.68	98.21	92.36	95.48
24	98.09	93.47	95.95	97.85	91.25	94.77	98.11	87.80	93.29	98.20	92.38	95.48
21	98.09	93.46	95.94	97.89	91.20	94.77	98.12	87.03	92.94	98.20	92.33	95.46
18	98.09	93.41	95.93	97.95	91.17	94.79	98.13	85.53	92.25	98.20	92.28	95.43
15	98.11	93.33	95.90	97.87	91.39	94.85	98.21	82.68	90.96	98.21	92.13	95.37
12	98.04	93.10	95.75	97.76	90.31	94.29	98.35	74.72	87.32	98.11	91.86	95.19
9	98.02	92.65	95.53	97.48	88.75	93.41	98.34	53.19	77.27	98.00	90.46	94.48
6	97.65	90.96	94.55	97.61	84.59	91.54	99.19	18.14	61.40	97.59	86.91	92.61
3	97.93	81.61	90.36	98.05	70.89	85.38	99.99	0.40	53.56	98.19	69.54	84.83
0	97.51	61.84	80.97	99.05	34.39	68.90	100.00	0.01	53.38	99.03	34.48	68.94
−3	98.35	31.50	67.38	99.99	0.57	53.63	100.00	0.00	53.37	99.96	0.60	53.63

Table 3.10 The result of Logistic regression models on LTAFDB dataset with and without adding noise (voting of K models)

Adding noise (dB)	jqrs			gqrs			wqrs			Voting (QRS)		
	Se	Sp	Acc	Se	Sp	Acc	Se	Sp	Acc	Se	Sp	Acc
Non	98.27	93.34	95.99	98.03	91.31	94.89	98.29	90.68	94.74	98.41	92.25	95.53
24	98.28	93.34	96.00	98.06	91.16	94.84	98.32	87.62	93.32	98.40	92.24	95.52
21	98.29	93.34	96.00	98.09	91.11	94.83	98.34	86.87	92.98	98.40	92.22	95.51
18	98.30	93.30	95.99	98.16	91.08	94.86	98.35	85.40	92.30	98.42	92.14	95.49
15	98.34	93.19	95.96	98.07	91.49	95.00	98.42	82.53	91.00	98.43	91.96	95.41
12	98.26	92.97	95.81	97.99	90.35	94.43	98.55	74.66	87.40	98.34	91.84	95.30
9	98.24	92.51	95.58	97.68	88.80	93.54	98.48	53.20	77.35	98.25	90.43	94.60
6	97.86	90.85	94.61	97.73	84.60	91.61	99.23	18.12	61.41	97.75	86.90	92.69
3	98.15	81.64	90.50	98.16	70.87	85.43	100.00	0.40	53.56	98.31	69.49	84.87
0	97.76	61.64	81.01	99.11	34.36	68.92	100.00	0.01	53.38	99.07	34.47	68.95
−3	98.55	31.21	67.35	99.99	0.57	53.63	100.00	0.00	53.37	99.97	0.59	53.63

without dropping, since both the QRS detectors which were used for bSQI cannot report correct QRS detections at the severe noisy situation.

Three QRS detectors and the majority voting of the three were used to analyze the LTAFDB dataset with and without adding noise. AF features were extracted from the RR intervals and were fed to the models which were established from the model development phase.

The result of LR models is shown in Tables 3.9 and 3.10 and Figures 3.2–3.4.

The result of SVM models is shown in Tables 3.11 and 3.12 and Figures 3.5–3.7.

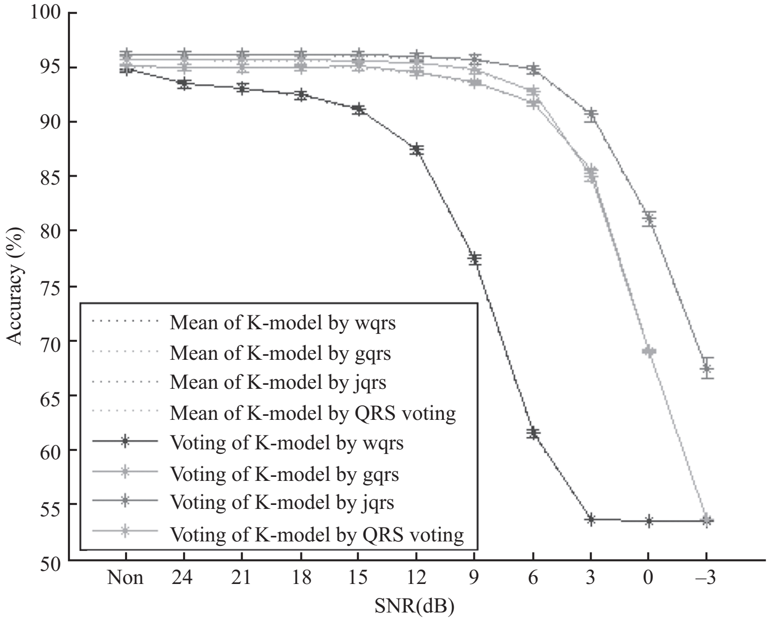

Figure 3.2 Accuracy of BLR models on LTAFDB dataset with adding noise

AQ9

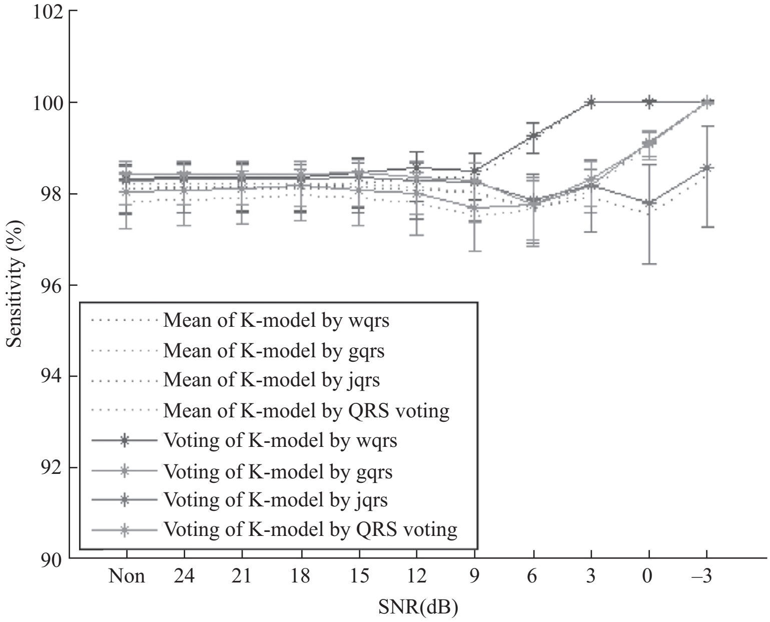

Figure 3.3 Sensitivity of BLR models on LTAFDB dataset with adding noise

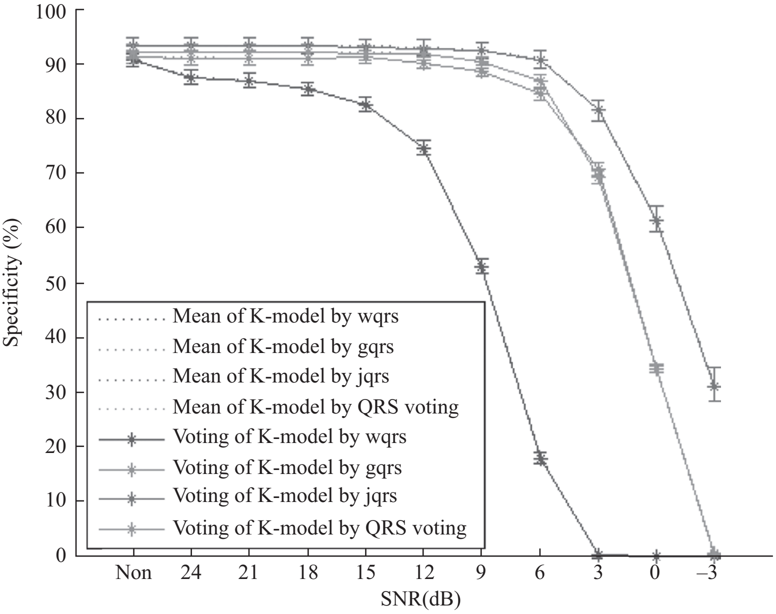

Figure 3.4 Specificity of BLR models on LTAFDB dataset with adding noise

Table 3.11 The result of SVM models on LTAFDB dataset with and without adding noise (mean of K models)

Adding noise (dB)	**jqrs**			**gqrs**			**wqrs**			**Voting (QRS)**		
	Se	**Sp**	**Acc**	**Se**	**Sp**	**Acc**	**Se**	**Sp**	**Acc**	**Se**	**Sp**	**Acc**
Non	96.29	95.43	95.89	96.45	93.27	94.79	96.67	93.23	94.85	96.76	94.79	95.84
24	96.29	95.43	95.89	96.51	93.11	94.75	96.65	91.58	94.08	96.79	94.97	95.94
21	96.28	95.41	95.88	96.51	93.05	94.73	96.67	90.72	93.69	96.77	94.84	95.87
18	96.27	95.36	95.85	96.56	92.99	94.73	96.66	89.18	92.96	96.75	94.81	95.85
15	96.21	95.38	95.82	96.39	92.71	94.52	96.67	86.66	91.79	96.70	94.80	95.82
12	95.99	95.27	95.65	96.13	91.68	93.87	96.66	79.01	88.10	96.39	94.41	95.46
9	95.57	95.08	95.34	95.36	90.04	92.71	95.95	59.44	78.25	95.56	92.92	94.33
6	94.79	94.13	94.49	95.33	87.16	91.44	97.08	25.93	62.09	94.54	91.37	93.06
3	94.37	87.17	91.03	95.78	73.72	85.41	99.79	6.83	52.93	94.43	76.43	86.04
0	92.22	72.04	82.86	97.32	37.06	69.12	99.79	9.14	52.48	95.48	41.73	70.42
−3	91.11	46.00	70.21	99.86	0.91	53.64	99.59	11.23	52.46	98.95	2.34	53.90

Table 3.12 The result of SVM models on LTAFDB dataset with and without adding noise (voting of K models)

Adding noise (dB)	**jqrs**			**gqrs**			**wqrs**			**Voting (QRS)**		
	Se	**Sp**	**Acc**	**Se**	**Sp**	**Acc**	**Se**	**Sp**	**Acc**	**Se**	**Sp**	**Acc**
Non	96.69	95.36	96.07	96.45	93.20	94.93	96.67	93.14	95.02	97.13	94.71	96.00
24	96.68	95.32	96.05	96.51	93.02	94.88	96.65	91.52	94.25	97.16	94.88	96.09
21	96.68	95.31	96.04	96.51	92.95	94.85	96.67	90.53	93.81	97.15	94.75	96.02
18	96.66	95.24	96.00	96.56	92.89	94.85	96.66	88.82	93.00	97.12	94.71	96.00
15	96.61	95.26	95.99	96.39	92.63	94.63	96.67	86.17	91.77	97.07	94.70	95.96
12	96.42	95.16	95.84	96.13	91.56	93.99	96.66	78.31	88.10	96.77	94.34	95.64
9	96.08	94.98	95.57	95.36	89.96	92.84	95.95	57.75	78.12	96.00	92.83	94.52
6	95.27	94.07	94.71	95.33	87.07	91.48	97.08	22.55	62.33	94.83	91.37	93.22
3	94.93	87.00	91.25	95.78	73.69	85.48	99.79	1.03	53.74	94.74	76.23	86.11
0	93.08	71.71	83.17	97.32	37.01	69.20	99.79	0.22	53.36	95.89	41.30	70.44
−3	92.21	45.41	70.53	99.86	0.78	53.66	99.59	0.35	53.32	99.17	1.92	53.83

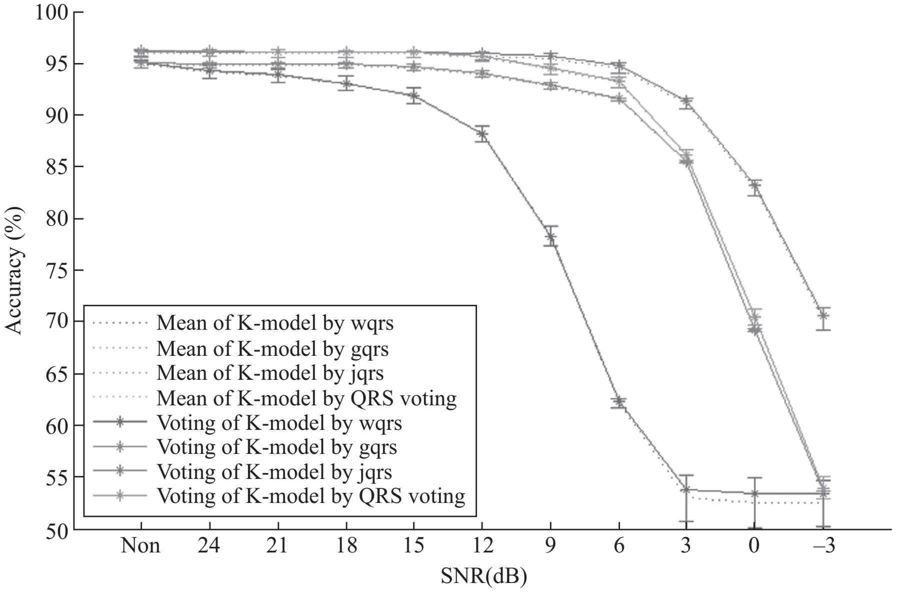

Figure 3.5 Accuracy of SVM models on LTAFDB dataset with adding noise

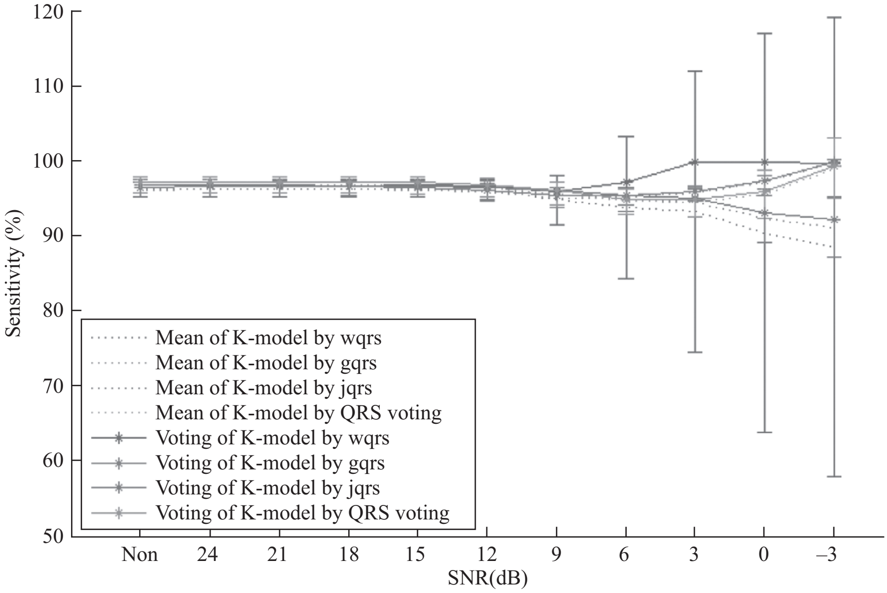

Figure 3.6 Sensitivity of SVM models on LTAFDB dataset with adding noise

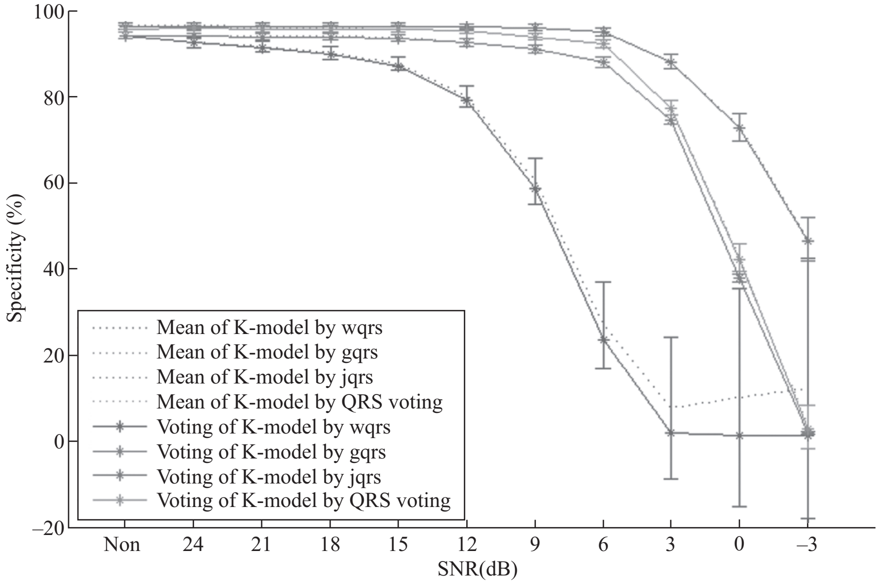

Figure 3.7 Specificity of SVM models on LTAFDB dataset with adding noise

3.7 Discussion

Note that each of the K-folds gives similar results and therefore it is hard to select which K-fold is likely to give better results on the test set. (Although we report the performance on each test set, this information should not be used to select a model, because it now becomes an intermediate validation set, and more held out data are required to evaluate the actual out-of-sample performance). There are several ways to deal with this, but essentially they boil down to a voting (and bagging) approach and using the out-of-bag error for estimating performance.

Unfortunately, in the absence of a new test sample, many researchers simply reapply the learning algorithm to the whole training set once the optimal cross-validated parameters have been found. However, overfitting is still possible since the data have already been used for model optimization. Alternatively, embedded methods, which allow feature ranking through a measure of variable importance, can be used. A typical example of this is Random Forests (RF), which is a form of bagging (although we have not explored RFs in this work).

In the examples we have presented here, we have taken the simplest approach and voted together each model developed on each fold together and cited results on the test set. In this case we find that we observe a modest rise in accuracy on the held out data from 96.9% to 98.1%. For more complex or nonlinear classifiers we may see larger improvements. However, there are also better ways to aggregate or vote together different classifiers, learning the physiological context in which each algorithm performs the best [33–35].

Finally we note that the use or real QRS detectors will result in errors in beat identification, even in low noise conditions. In reality, the noise can be extremely high from ambulatory activity, and so rejection of noisy segments needs to be considered very carefully. We refer the reader to Oster and Clifford [36] for more details on this subject. In this work, three popular QRS complex detection methods were evaluated on ECG signals with different SNRs. "jqrs" method [11,12] consists of a window-based peak energy detector and essentially also a Pan and Tompkins (P&T)-like QRS detector [37]. Compared with the P&T method, it used a smaller window size (27 ms vs. 100 ms) thus inducing a better performance for rejecting the false detection due to the high amplitude T waves. In the "jqrs" method, the original band-pass filter was replaced with Mexican hat filter and an additional heuristic ensuring no detection was attempted during very low amplitude unvarying ECG (flat lines). A search-back procedure is also allowed in case of suspected missed beats. This combination provided the best performance among the three selected QRS complex detection methods. The "gqrs" method consists of a QRS matched filter with a custom built set of heuristics (such as search back). Unfortunately this method does not have an associated publication. So it is hard to comprehensively explain the implementation. The "wqrs" method [13] involves low-pass filtering of the ECG followed by a nonlinearly scaled curve length transformation and a series of decision rules. This method had the lowest performance of the three detectors on LTAFDB dataset, especially when the ECG signals were contaminated by realistic noise. From Figures 3.2–3.7, it can be easily seen that with the increase of SNR values, accuracy and specificity values of the "wqrs" method dropped rapidly. Although its sensitivity did not drop greatly, the standard deviations became much larger than "jqrs" and "gqrs" methods. We also note that one may expect that the majority voting method would be expected to report better results than any of the independent QRS complex detection methods. From Tables 3.9–3.12, we can see that the voting method usually reported worse performances than "jqrs" method but better performance than other two independent methods. This may be because "wqrs" and "gqrs" respond to artifacts in a similar manner and are not truly independent. In fact, we have shown in earlier works that voting methods only provides substantial improvements over the best algorithm if each detection (or vote) is weighted based on the relative performance of the algorithm, particularly in the context of physiology and noise. For more details we refer the reader to Zhu *et al.* [33].

In conclusion we emphasize the following points:

1. Most literature reports over-trained data, and uses small numbers of patients drawn from a single database. Testing on completely unseen databases is required to provide some level of trust in the signal.
2. Most databases are handpicked to be clean. Testing on such data misrepresents the performance of an algorithm in the real world. Realistic noise should be titrated into the data and the performance of a classifier be tested as a function of such noise. (White and stationary noise is an unacceptable test.)
3. Signal quality metrics are important for identifying noisy periods of data and rejecting them from classification, or for allowing a classifier to learn the class output in the context of such noise. They also provide objective ways to assess the confidence intervals on the classifier's output.
4. Many databases contain expert annotations. Training and testing on these leads to an overly optimistic result. When automated algorithms are used to identify the features to present to a classifier, significant drops in performance are observed.
5. Voting together classifiers or detectors improves the output, but generally only if you have large numbers of them, and/or can weight them using context (such as physiology and/or signal quality).

Appendix 1

Coefficient of sample entropy (COSEn)

COSEn was defined by Lake *et al.* [2,3] as an entropy measure derived from SampEn, designed specifically to detect AF in very short RR time series [2,3]. To avoid the less confident entropy estimates because of falling numbers of matches of length m and matches of length $m + 1$ due to the relatively small fixed r values, a measure called quadratic sample entropy (QSE), based on densities rather than probability estimates, was introduced in Reference 7. It normalized SampEn by the volume of each matching region, i.e., $(2r)^m$:

$$\begin{aligned}\mathrm{QSE} &= -\ln\left(\frac{A^{m+1}(r)/(2r)^{m+1}}{B^{m}(r)/(2r)^{m}}\right) = -\ln\left(\frac{A^{m+1}(r)}{B^{m}(r)}\right) + \ln(2r)\\ &= \mathrm{SampEn} + \ln(2r) \qquad \text{(A1)}\end{aligned}$$

In addition, regression analyses showed that heart rate was an important independent predictor of AF [3]. Hence, the COSEn measure uses the concept of density estimates of QSE but subtracts the natural logarithm of the mean RR interval from QSE as:

$$\mathrm{COSEn} = \mathrm{SampEn} + \ln(2r) - \ln(\mathrm{mean(RR)}) \qquad \text{(A2)}$$

where both r and mean(RR) use the unit of s.

Normalized fuzzy entropy (NFEn)

First, we generated quadratic fuzzy local measure entropy (QFLMEn) and quadratic fuzzy global measure entropy (QFGMEn) based on the density estimates rather than probability estimates by normalizing the FLMEn and FGMEn using the volume of each matching region, i.e., $(2r)^m$:

$$\begin{aligned}\mathrm{QFLMEn} &= -\ln\left(\frac{AL^{m+1}(n_L,r_L)/(2r)^{m+1}}{BL^{m}(n_L,r_L)/(2r)^{m}}\right) = -\ln\left(\frac{AL^{m+1}(n_L,r_L)}{BL^{m}(n_L,r_L)}\right) + \ln(2r)\\ &= \mathrm{FLMEn} + \ln(2r)\\ \mathrm{QFGMEn} &= -\ln\left(\frac{AG^{m+1}(n_G,r_G)/(2r)^{m+1}}{BG^{m}(n_G,r_G)/(2r)^{m}}\right) = -\ln\left(\frac{AG^{m+1}(n_G,r_G)}{BG^{m}(n_G,r_G)}\right) + \ln(2r)\\ &= \mathrm{FGMEn} + \ln(2r) \qquad \text{(A3)}\end{aligned}$$

We also used the conclusion of “regression analyses showed that heart rate was an important independent predictor of AF” in Reference 3, and subtracted the natural logarithm of the mean RR interval from QFLMEn and QFGMEn as:

$$\begin{aligned}\mathrm{QFLMEn} &= \mathrm{FLMEn} + \ln(2r) - \ln(\mathrm{mean(RR)})\\ \mathrm{QFGMEn} &= \mathrm{FGMEn} + \ln(2r) - \ln(\mathrm{mean(RR)}) \qquad \text{(A4)}\end{aligned}$$

And finally, NFEn is calculated as:

$$\begin{aligned}\mathrm{NFEn} &= \mathrm{QFLMEn} + \mathrm{QFGMEn}\\ &= \mathrm{FLMEn} + \mathrm{FGMEn} + 2 \times \ln(2r) - 2 \times \ln(\mathrm{mean(RR)})\\ &= \mathrm{FuzzyMEn} + 2 \times \ln(2r) - 2 \times \ln(\mathrm{mean(RR)}) \qquad \text{(A5)}\end{aligned}$$

References

[1] Clifford G.D., Long W.J., Moody G.B. and Szolovits P. Robust parameter extraction for decision support using multimodal intensive care data. *Philosophical Transactions of the Royal Society A*, 2009; 367(1887): 411–429.

[2] Clifford G.D., Behar J., Li Q. and Rezek I. Signal quality indices and data fusion for determining acceptability of electrocardiograms collected in noisy ambulatory environments. *Physiological Measurement,* 2012; 33: 1419–1433.

[3] Li Q. and Clifford G.D. Signal quality and data fusion for false alarm reduction in the intensive care unit. *Journal of Electrocardiology*, 2012; 45: 596–603.

[4] Fraser H.S. and Joaquin B. Implementing medical information systems in developing countries, what works and what doesn't. *AMIA Annual Symposium Proceedings*, 2010; 232–236.

[5] Gerber T., Olazabal V., Brown K. and Pablos-Mendez A. An agenda for action on global e-health. *Health Affairs*, 2010; 29: 233–236.

[6] Waegemann C.P. mHealth: the next generation of telemedicine? *Telemedicine Journal e-Health*, 2010; 16: 23–25.

[7] Tamrat T. and Kachnowski S. Special delivery: an analysis of mHealth in maternal and newborn health programs and their outcomes around the world. *Maternal and Child Health Journal*, 2012 Jul; 16(5): 1092–1101.

[8] Li Q., Mark R.G. and Clifford G.D. Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter. *Physiological Measurement,* 2008 Jan; 29(1): 15–32.

[9] Zong W., Moody G.B. and Mark R.G. Reduction of false arterial blood pressure alarms using signal quality assessment and relationships between the electrocardiogram and arterial blood pressure. *Medical and Biological Engineering & Computing,* 2004; 42: 698–706.

[10] Fuster V., Ryden L.E., Cannom D.S., *et al.* ACC/AHA/ESC 2006 guidelines for the management of patients with atrial fibrillation. *Circulation*, 2006; 8(9): 651–745.

[11] Behar J., Johnson A., Clifford G.D. and Oster J. A comparison of single channel fetal ECG extraction methods. *Annals of Biomedical Engineering*, 2014; 42: 1340–1353.

[12] Behar J., Oster J. and Clifford G.D. Combining and benchmarking methods of fetal ECG extraction without maternal or scalp electrode data. *Physiological Measurement*, 2014; 35: 1569.

[13] Zong W., Heldt T., Moody G. and Mark R. An open-source algorithm to detect onset of arterial blood pressure pulses. *Proceedings Computers in Cardiology*, 2003; 259–262.

[14] Behar J., Oster J., Li Q. and Clifford G.D. ECG signal quality during arrhythmia and its application to false alarm reduction. *IEEE Transactions on Biomedical Engineering*, 2013; 60(6): 1660–1666.

[15] Li Q., Rajagopalan C. and Clifford G.D. A machine learning approach to multilevel ECG signal quality classification. *Computer Methods and Programs in Biomedicine*, 2014 Dec; 117(3): 435–447.

[16] Carrara M., Carozzi, L., Moss, T.J., *et al.* Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy. *Physiological Measurements*, 2015; 36(9): 1873–1888.

[17] Behar, J., Andreotti, F., Zaunseder, S., Li, Q., Oster, J. and Clifford, G.D. An ECG simulator for generating maternal-foetal activity mixtures on abdominal ECG recordings. *Physiological Measurement,* 2014; 35(8): 1537.

[18] Corino V.D., Sandberg F., Mainardi, L.T. and Sornmo, L. An atrioventricular node model for analysis of the ventricular response during atrial fibrillation. *Biomedical Engineering, IEEE Transactions* on, 2011; 58(12): 3386–3395.

[19] Lake D.E. and Moorman J.R. Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted ventricular devices. *American Journal of Physiology-Heart and Circulatory Physiology*, 2011; 300(1): H319–H325.

[20] Sarkar S., Ritscher D. and Mehra R. A detector for a chronic implantable atrial tachyarrhythmia monitor. *IEEE Transactions on Biomedical Engineering*, 2008; 55(3): 1219–1224.

[21] Colloca R., Johnson A.E., Mainardi L. and Clifford G.D. A support vector machine approach for reliable detection of atrial fibrillation events. In *Computing in Cardiology Conference (CinC*). New York: IEEE, 2013 (pp. 1047–1050).

[22] Task Force of the European Society of Cardiology. Heart rate variability standards of measurement, physiological interpretation, and clinical use, *European Heart Journal,* 1996; 17: 354–381.

[23] Carrara M., Carozzi L., Moss T.J. *et al.* Heart rate dynamics distinguish among atrial fibrillation, normal sinus rhythm and sinus rhythm with frequent ectopy. *Physiological Measurements,* 2015; 36(9): 1873–1888.

[24] Lake D.E. and Moorman J.R. Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted ventricular devices. *American Journal of Physiology Heart and Circulatory Physiology*, 2011; 300(1): H319–H325.

[25] Liu, C.Y., *et. al.*, Comparison of different entropy measures for atrial fibrillation detection, ready to submit. AQ10

[26] D.T. Linker. Long-term monitoring for detection of atrial fibrillation. US Patent 7630756 B2, 2009. AQ11

[27] Sarkar S., Ritscher D. and Mehra R. A detector for a chronic implantable atrial tachyarrhythmia monitor. *IEEE Transactions on Biomedical Engineering*, 2008; 55(3): 1219–1224.

[28] Lake D.E. Renyi entropy measures of heart rate Gaussianity. *IEEE Transactions on Biomedical Engineering*, 2006; 53(1): 21–27.

[29] Guyon I., Gunn S., Nikravesh M. and Zadeh L.A. *Feature Extraction – Foundations and Applications*. Berlin Heidelberg: Springer, 2006.

[30] Boser B.E., Guyon I. and Vapnik V. A training algorithm for optimal margin classifiers. *Proceedings of Fifth Annual Workshop on Computational Learning Theory.* ACM, 1992: 144–152.

[31] Schölkopf B. and Smola A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge: MIT Press; 2001.

[32] Guyon I., Weston J., Barnhill S. and Vapnik V. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 2002; 46: 389–422.

[33] Zhu T., Johnson A.E., Behar J. and Clifford G.D. Crowd-sourced annotation of ECG signals using contextual information. *Annals of Biomedical Engineering,* 2014 Apr; 42(4): 871–884.

[34] Zhu T., Pimentel M.A.F., Clifford G.D. and Clifton, D.A. Bayesian fusion of algorithms for the robust estimation of respiratory rate from the photoplethysmogram. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015: 6138–6141. doi:10.1109/EMBC.2015.7319793

[35] Zhu T., Dunkley N., Behar J. Clifton D.A. and Clifford, G.D. Fusing continuous-valued medical labels using a bayesian model. *Annals of Biomedical Engineering*, 2015; 43(12): 2892–2902. doi:10.1007/s10439-015-1344-1.

[36] Oster J. and Clifford G.D. Impact of the presence of noise on RR interval-based atrial fibrillation detection. *Journal of Electrocardiology*. 2015; 48(6): 947–951.

[37] Pan J. and Tompkins W.J. A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*. 1985; 32: 230–236.

Chapter 3

Signal processing and feature selection preprocessing for classification in noisy healthcare data

Author Queries

AQ1: Please confirm if the hierarchy of section headings is fine.

AQ2: Please expand the term “ABP” and confirm whether it is necessary.

AQ3: We have shortened running head in recto pages. Please check and confirm.

AQ4: Please confirm if the edits made to the sentence “Here we selected a 30 …. point of view” are fine.

AQ5: Please expand the term “SNR.”

AQ6: Please expand the term “MAD.”

AQ7: “Se”, “Sp” and “Acc” are found both Romanized and italicized in the chapter. Please confirm if this is intentional.

AQ8: Please provide the significance of the bolded text in Table 3.6.

AQ9: The book will be printed using black only so any colour images will need to be converted to greyscale. Kindly resubmit in greyscale any colour figures that won’t make sense once converted.

AQ10: Per style, all authors of each work should be listed, up to a maximum of six. If there are more than six authors only the first three should appear, followed by *et al*. Hence kindly provide the complete author group for Reference 25.

AQ11: Please update the reference.