Transfer Report

Yanting Shen
Trinity College
University of Oxford
Supervised by: Professor David Clifton
Professor Robert Clarke
Professor Zhengming Chen
25 September 2016
Abstract

We set out to use machine learning techniques to analyse ECG data to improve risk evaluation of cardiovascular disease in a very large cohort study of the Chinese population. We performed this investigation by (i) detecting “abnormality” using 3 one-class classification methods, and (ii) predicting probabilities of “normality”, arrhythmia, ischemia, and hypertrophy using a multiclass approach.

For one-class classification, we considered 5 possible definitions for “normality” and used 10 automatically-extracted ECG features along with 4 blood pressure features. The one-class approach was able to identify abnormality with area-under-curve (AUC) 0.83, and with 75.6% accuracy.

For four-class classification, we used 86 features in total, with 72 additional features extracted from the ECG. Accuracy for this four-class classifier reached 75.1%. The methods demonstrated proof-of-principle that cardiac abnormality can be detected using machine learning in a large cohort study.

Our work contributes to the existing field of ECG risk evaluation by formulating a four-class model for clinical integration of time-series data to electronic data records (EHR); proposing a framework to reweight the posterior for an imbalanced training set; and proposing a framework to evaluate the model against several “silver” standards when the “gold” standard is absent.

Related Publication

# Table of Contents

Abstract................................................................................................................................. 2
Related Publication................................................................................................................ 2

1. INTRODUCTION...................................................................................................................... 4

2. MACHINE LEARNING FOR ECG RISK EVALUATION: EXISTING STUDIES .... 6
   2.1 Feature extraction models.............................................................................................. 6
   2.2 Classification.................................................................................................................. 8
   2.3 Other Machine learning models for time-series analysis .............................................. 12
   2.4 Conclusion ................................................................................................................... 13

3. DATASET DESCRIPTION........................................................................................................ 15
   3.1 ECG time-series ......................................................................................................... 15
   3.2 ECG features ............................................................................................................ 15
   3.3 Blood pressure data ................................................................................................... 16
   3.4 Textual labels ........................................................................................................... 16
   3.5 Health Insurance Data ............................................................................................. 18

4. PREPROCESSING AND VISUALISATION............................................................................ 19
   4.1 Feature Selection......................................................................................................... 19
   4.2 Visualisation ............................................................................................................... 20
   4.3 Signal Quality Evaluation ......................................................................................... 21
   4.4 Feature Extraction.................................................................................................... 21

5. ANALYSIS ............................................................................................................................ 22
   5.1 One-class classification ............................................................................................ 22
   5.2 Four-class classification ......................................................................................... 26
   5.3 Results and Discussion ............................................................................................ 27

6. CONCLUSIONS AND FUTURE WORK .......................................................................... 31
   6.1 Summary of work to date ......................................................................................... 31
   6.2 Immediate future work ............................................................................................. 33
   6.3 Plan until completion............................................................................................... 39
   6.4 Risk assessment....................................................................................................... 40

Appendix.................................................................................................................................. 42
   Appendix A: Gaussian Process Regression ....................................................................... 42
   Appendix B: The Gaussian Process Latent Variable Model .............................................. 46
   Appendix C: Support Vector Machines ............................................................................ 48

Bibliography........................................................................................................................... 52
1. INTRODUCTION

Cardiovascular disease (CVD), including coronary heart disease, cerebrovascular disease, peripheral arterial disease, rheumatic heart disease, congenital heart disease, deep vein thrombosis, and pulmonary embolism, is the leading cause of mortality worldwide [1] and in China [2]. An estimated 3.76 million people died from CVD in China in 2013 [1]. There are large geographical and economic variations of CVD mortality in China [2], suggesting appropriate measures may be taken for prevention and effective treatment of the disease. For example, the World Health Organisation advises people with high CVD risk to access early detection and management facilities [1]. Studies have shown effective reduction of CVD mortality rates by promoting awareness of risk factors and adopting lifestyle changes [1]. Identifying risks to the Chinese population will help provide advice to people to improve their lifestyle and help clinicians to discover appropriate treatments for specific conditions; this promises to reduce mortality and healthcare expenditure. Many CVD risk factors have been identified by long-term follow-up studies, such as obesity [3-5], diabetes [6], metabolic syndrome [7], smoking [1], and hypertension [8], among many others. Recently, multiple biochemical and genetic risk factors have been identified via conventional follow-up approaches, and by machine learning in large datasets [9].

While much research has been focusing on the effects of certain risk factors, personalised medicine requires integrating all risk factors to which a person is exposed and then predicting risks for specific diseases, so that preventative measures may be taken. Examples of research for this purpose include [10], in which mathematical models were established by scoring systems using multiple risk factors (blood pressure, total cholesterol, high-density lipoprotein cholesterol, smoking, glucose intolerance, and left ventricular hypertrophy) as inputs to predict the risks of
myocardial infarction, coronary heart disease, CVD, and death from these diseases. With an increasing number of risk factors being identified, and especially with abundant genetic and lifestyle data now available, it can be expected that such an approach will face difficulty as the “healthy” range of the newly-identified factors are hard to obtain or quantify.

Machine learning has the advantage of estimating the associations between risk factors and diseases without prior knowledge of accurate reference values of the risk factors. This approach has grown popular in risk evaluation and diagnostics for chronic diseases [11-14]. In CVD, Knuiman et al. have predicted coronary mortality in the Busselton cohort using a discriminative decision tree [15]; Lapuerta et al. used a neural network in the prediction of coronary disease risk from serum and lipid profiles [16]; and Das et al. performed heart disease diagnosis using ensembles of neural networks [17].

The electrocardiogram (ECG), being an important measurement of cardiac function that is relatively easy to obtain, is surprisingly seldom used in prediction. This report attempts to address the need for risk metrics that include ECG-derived features by analysing CVD risks associated with abnormal ECG, using the China Kadoorie Biobank ECG dataset. This report includes two risk evaluation tasks: (i) “abnormality” detection and (ii) prediction of probabilities of “normal”, “arrhythmia”, and “ischemia”. Since abnormality is relatively rare in this database, novelty detection (the aim of which is to classify an under-sampled “abnormal” class) is an appropriate approach to address the first task. To address the second task we build models of “normality”, “arrhythmia”, and “ischemia” on which probabilistic prediction of the “borderline” data will be based.

This report includes a summary of existing studies on machine learning models for ECG
risk evaluation, followed by analysis of the two risk evaluation tasks. For the first task we compare prediction performance of three one-class algorithms: (i) a generative KDE, (ii) a discriminative KDE, and (iii) a discriminative SVM. They will be compared under five different assumptions for the “normal” class. For the second task we propose a set of inclusion criteria for defining the four classes corresponding to “normality”, “arrhythmia”, “ischemia”, “hypertrophy”, and present classification accuracy using a support vector regression (SVR) with additionally extracted features. Finally, the 4-class model was used to predict the “borderline” data which are not assigned class membership, and compared with disease endpoints from the Health Insurance (HI) dataset.

2. MACHINE LEARNING FOR ECG RISK EVALUATION: EXISTING STUDIES

The typical 3-step framework for machine learning risk analysis with ECG data is 1) feature extraction, 2) classification, and 3) model evaluation [9]. Risk evaluation tasks are usually evaluated by how accurately the model predicts the labels that are used as the “gold standard”. Examples of gold standards include human experts, and standard databases such as the MIT-BIH arrhythmia database. Therefore, the literature involving risk analysis of ECG data is often referred to as ECG classification, and the aim is to increase classification accuracy, sensitivity, and specificity with respect to the gold standard.

2.1 Feature extraction models

For feature extraction, the most commonly used algorithms include the wavelet transform [17-19], genetic algorithms for searching for candidate features [17], dynamic time warping (DTW) [20, 21], principle component analysis (PCA) [22], adaptions of the k-nearest neighbour (kNN)
algorithm for feature extraction [20], symbolic aggregate approximation (SAX) [20], correlation-based feature selection [23], linear forward selection [23], power-spectrum methods [22], Lyapunov exponents [18], fractal dimension [18], and morphology filtering [22], among many others.

A classic example for improving classification accuracy is to discover novel patterns in time-series [20], where Syed et al. improved risk stratification after acute coronary syndrome by 7 to 13% using three computational ECG biomarkers: morphologic variability (MV), symbolic mismatch (SM), and heart rate motifs (HRM). MV, which quantifies the energy difference between consecutive heart beats, was extracted by a process that included dynamic time warping to remove the effect of time distortions in the ECG caused by respiration and other contributing physiological factors. SM which measures the inter-patient difference was modelled via kNN novelty detection after iterating a max- min clustering algorithm with a DTW distance metric. HRM was extracted by symbolic aggregate approximation (SAX). These features are all unintuitive to human inspection, but which were shown to be useful indicators of risk using the ECG.

Kim et al. proposed a robust arrhythmia classification algorithm using an extreme learning machine, and used morphology filtering combined with PCA to classify 6 beat types [22]. Ye et al. used morphological and dynamical features based on the wavelet transform and independent component analysis (ICA) for classification of 16-class ECG beats [24]. They built decision systems involving the fusion of results from two ECG leads, and reported 99.3% classification accuracy.

El-Dahshan et al. used a hybrid involving genetic algorithms and wavelets to denoise the ECG [17]. Goletsis et al. classified ischemic beats with a genetic algorithm which thresholds five
criteria describing ST segment changes, T wave alternans, and the patient’s age [25]. Their sensitivity and specificity were both above 90%.

Arif et al. used the discrete wavelet transform to classify 6 beat types from ECG data with 99.5% accuracy, by using a KNN [26]; this work used PCA for feature reduction. Povielli et al. proposed a novel approach that represents time-series data by a reconstructed phase space, and tested the resulting feature space for use with arrhythmia and speech reorganization datasets; this work used artificial neural network approaches [27].

2.2 Classification

For classification, the most commonly used classifiers include the support vector machine (SVM, Appendix C) [19, 21, 22, 28-34], the multi-layer perceptron (MLP) [21-23, 34-36], the extreme learning machine (ELM)[18, 22, 29], random forests [34, 37], k-nearest neighbour (kNN) [21], Bayesian decision making (BDM) [21], other artificial neural networks (ANNs) including aspects of least-square methods (LSM) [21], polynomial classifiers [38], ensembles [39], fuzzy finite state machines [40], stochastic Petri nets [40], decision trees [41], evolution algorithms [18], self-organising maps [35], radial basis function networks [22], and straightforward linear regression [34].

For example, Mitra et al. detected arrhythmia from ECG data using an incremental back propagation neural network (IBPLN) and reported accuracy up to 87.7% [23]. They selected features using correlation-based methods and linear forward selection. Fergus et al. predicted preterm delivery of infants using a polynomial classifier on ECG and achieved 96% sensitivity, 90% specificity, and 95% AUC [38]. Tantimongcolwat et al. used a back-propagation neural network (BNN) and a direct kernel self-organising map (DK-SOM) for detection of ischemic heart
disease on magneto-cardiograms (MCG), and achieved accuracy of to 80.4% on 125 individuals [35]. Sun et al. achieved high performance in ECG classification with a reusable neuron architecture (RNA) [36].

Kurz et al. developed a simple point-of-care risk stratification for acute coronary syndrome (ACS) called “Acute Myocardial Infarction in Switzerland” (AMIS), and tested on 7520 participants. The study reported that the AMIS model outperformed several clinically-accepted risk score systems [41]. Their classifier was based on Average One-Dependence Estimator (AODE) and a J48 decision tree with a sequential backward deletion process for feature selection.

Hsich et al. used random survival forests to identify important risk factors for survival in patients with systolic heart failure and achieved similar accuracy to a Cox proportional hazard regression model; they found that the random forest offered more intuitive illustration of important risk factors (and interactions between multiple risk factors) [37].

Arsanjani et al. used an ensemble-boosting algorithm to classify myocardial perfusion after single- photon emission computed tomography (SPECT) for coronary artery disease in a large population (n = 1181), and compared results with those from a clinically-used risk score involving total stress perfusion deficit (TPD) and two human experts [39]. Their results outperformed TPD and human experts in terms of ROC (0.94) and achieved at least as good sensitivity, specificity, and accuracy.

2.2.1 The SVM as a robust classifier

The SVM has been shown to be particularly robust in many settings and has thus been used in clinical studies either as the only classifier or as a comparison with the proposed algorithm. Its mathematical formulation can be found at Appendix C. For example, Moavenin et al. improved
the basic SVM with a kernel-Adatron algorithm and compared to results from a MLP in the classification of 6 arrhythmia beat types [42]. It was concluded that the SVM significantly outperformed the MLP. Khorrani et al. used continuous wavelet transforms, discrete wavelet transforms, and discrete cosine transforms for feature extraction; they performed arrhythmia classification with an MLP and a kernel-Adaptron SVM using two-lead ECG [43]. This study also confirmed that the SVM outperformed the MLP. Melgani et al. demonstrated the general superiority of SVM compared to kNN and RBF networks in ECG classification, and improved the SVM algorithm by using particle swarm optimisation (PSO); their study reported 89.7% accuracy in the classification of 40,438 beats [44].

Similarly, Artis et al. conducted automated screening of arrhythmia using the discrete wavelet transform. They trained two class classifiers (normal and arrhythmia) on the MIT-BIH database and tested on a clinical cohort [45]. They reported that the highest accuracy was obtained by an SVM, compared to an MLP and a Gaussian mixture model (GMM).

An SVM was used to classify signal quality of ECG and EEG, and achieved 92% accuracy compared to human labels [30]. Khandoker et al. used an SVM to detect obstructive sleep apnea syndrome (OSAS) from 8-hour ECG recordings and achieved 92.85% accuracy [31]. Kampouraki et al. used an SVM for heart-beat time-series classification using long-term ECG recordings [32]. Bsoul et al. developed a real-time sleep apnea monitor, called “Apnea MedAssist”, which used an SVM, and reported a sensitivity of 96% [33]. The algorithm requires only 1-minute ECG recordings. Li et al. used an SVM to classify ventricular fibrillation and tachycardia classification with 14 features [28]. They used a genetic algorithm for feature selection, and achieved 96.3% accuracy, 96.2% sensitivity, and 96.2% specificity. Ucebeyli et al. classified normal and partial epilepsy from ECG using an SVM and achieved 99.44% accuracy, using wavelet coefficients as
features [19]. Yu et al. improved ICA for ECG beat classification by “independent component rearrangement” and achieved 98.7% accuracy in 8 beat types using a probabilistic neural network and a SVM [46]. Clifton et al. proposed a predicted and personalised multivariate early-warning system using a one-class SVM and proposed a new way to parametrize the method for clinical tasks using partial AUROC [47].

### 2.2.2 Classifiers outperforming SVMs

A number of models have been reported to outperform SVMs. For example, Özdemir et al. used six machine learning classifiers (KNNs, LSMs, SVMs, Bayesian decision marking, dynamic time warping, and ANNs) to detect falls from gyroscope, accelerometer, and magnetometer data [21]. They tested on a set of 2,520 trials comprising 20 voluntary falls and 16 types of daily living activities. The best results were achieved by the KNN and LSM, with sensitivity, specificity, and accuracy all above 99%.

Karpagachelvi showed that an extreme learning machine was superior to an SVM in ECG classification because the former can optimise its parameters to best tune its discrimination of classes, as well as identifying the best subset of features for the task [29]. Compared with the SVM, another study concluded that methods such as MLPs, and RBF networks, and extreme learning machine were more accurate and faster [22] for their task.

Zavar et al. reported that the SVM was outperformed by an evolutionary model in seizure detection from ECG, in which feature extraction was performed via Lyapunov exponents, fractal dimension, and wavelet entropy [18]. Monte-Moreno et al. estimated blood glucose and blood pressure from the photoplethysmogram acquired from pulse oximeters with 410 participants using ridge linear regression, MLPs, SVMs, and random forests, where the best result was achieved by
the latter [34].

2.3 Other Machine learning models for time-series analysis

Machine learning is readily applied to medical monitoring with time-series vital signs. For example, Pullinger et al. implemented an automated system which calculates an early warning score (EWS), and after implementing the automated system, the percentage of patients being assigned an accurate EWS score increased from 52.7% to 92.9% [48]. In medical monitoring, changes in time-series vital signs often indicate changes in underlying disease condition. Such deviation from the “normal state” can be modelled by novelty detection approaches.

A Gaussian process (GP), which models the time-series as a distribution over functions, is a one of the methods that are especially suitable to model the “normal state” for time-series data, therefore we give a more detailed account here. The mathematical basics of GP can be found in Appendix A.

Dürichen et al. proposed a multitask Gaussian process that can learn the correlation between multiple vital signs [49]. The “abnormal state” can be classified by thresholding on the resulting predicted risk scores using probabilistic extreme value theory (EVT) [50]. For example, Clifton et al. generalised conventional univariate EVT for multivariate, multimodal problems [51], and further extended the Generalised Pareto Distribution (GPD), which is followed by the extreme values of any identical and independent distributed (i. i. d.) random variables from non-degenerate distributions, to apply to high-dimensional data [52].

In [53], the “abnormality” in time-series was detected in a Gaussian process framework using EVT. The functions from the output of a Gaussian process were mapped to probability
density \( z \), then a distribution \( f(z) \) was defined over those densities; finally, an extreme function distribution \( F(r) \) was used to characterise the extremes of \( G(z) \), where \( r \) is the Mahalanobis radius from the “normal” model \( M \). This approach showed the potential to classify “true” abnormality from “extreme-but-normal” examples in time-series physiological data.

The strengths of Gaussian process regression include 1) the fact that it is a non-parametric approach, and many other methods have been shown to relate to Gaussian process. For example, relevant vector machines can be presented as a special case of a GP [54]; 2) The property that the joint distribution is a Gaussian makes use of the convenient properties of Gaussian, i.e. the conditional and marginal distributions are also Gaussian, which enables analytical solution in many cases; 3) GP is intrinsically Bayesian, as shown in the inference of regression and classification (Appendix A); 4) GP makes use of the so-called “kernel trick” which reduces computational cost in many cases. An additional merit of GP for classification, compared to probabilistic SVMs (which transforms some linear transformation of the discrimination boundary through a sigmoid function) is that GP classification takes account into the predictive variance of the latent function assumed to generate the data, \( f(x) \). The main limitation of GP regression is that the basic computational complexity is \( O(n^3) \), due to the inversion of an \( n \) by \( n \) matrix (where \( n \) is the number of data), which therefore can be intractable for large datasets. Many approximation strategies have been proposed for using GPs with large datasets, such as reduced rank approximation of the Gram Matrix, greedy approximation, approximation with fixed hyperparameters, and approximate the marginal likelihood and its derivatives. Details of approximation algorithms can be found at chapter 8 of [54], and is omitted here.

2.4 Conclusion

There is a large body of literature concerning ECG classification, which reports high classification
accuracy, especially concerning arrhythmia classification. The SVM is reported as being one of the most commonly-used and robust classifiers for ECG-based analyses, and is often used as a point of comparison when a new model is proposed. Novelty detection approaches and algorithms exploiting Gaussian process regression may be applied to ECG signal analysis for characterizing the dynamical aspects of a time-series, which offers particular appeal in the setting of ECG analysis.

While these models yield high classification accuracy, they are mostly limited to a nearly-balanced, small sample size chosen from distinctive groups labelled by clinical experts, and using long ECG recordings. In comparison, the major challenges of our research include 1) a large sample size ($n = 25019$) and short ECG recording length ($t = 10s$); 2) noisy labels due to a lack of human expert labelling; 3) unbalanced classes with a large number of data points in the “middle ground” between different classes; 4) the substantial time elapsed between clinical outcome and ECG acquisition, as described in Chapter 3. These unique challenges pose the following research questions that this research aims to address:

- How should we integrate heterogeneous data types in the electronic health records (EHR), and in particular, how should we handle the time-series data in a robust manner?
- How should we handle datasets with heavily disproportional representation of classes?
- With the absence of the “gold” standard, and the presence of several “silver” standards (as is typical in clinical applications), how should we evaluate the models in a principled way?

In the following sections we will propose our solutions to the above questions, and, in the conclusion and future work section, we will discuss ways to further provide more precise solutions by using more advanced machine learning models.
3. DATASET DESCRIPTION

The China Kadoorie Biobank (CKB) is a prospective cohort study of over 520,000 adults from 10 areas in China during 2004-2008 [55]. Data were collected using questionnaires; physiological measurements were recorded at baseline and all participants provided a blood sample. Information on cause of death rates was collected from health insurance data, and from mortality and disease registries. After five years, approximately 25,000 surviving participants were resurveyed with further questionnaires, ECG measurements, and blood collection. We have institutional ethics approval to use the data. Public access to the CKB data can be found at http://www.ckbiobank.org/site/Data+Access.

The data available for our study include:

3.1 ECG time-series

Standard 12-lead ECG (10-s duration, 500Hz) was recorded on 24,369 participants using a Mortara ELIx50 device in 2013-2014. Also available is a “typical cycle” from each lead for each participant, which was generated by the device using a proprietary algorithm.

3.2 ECG features

The Mortara device provides 19 features (Table 2) which were automatically extracted from the “typical cycles” for each participant. A schematic representation of a selection of these features is shown in Figure 1.
3.3 Blood pressure data

Systolic blood pressure and diastolic blood pressure were recorded twice on each individual using standard methods.

3.4 Textual labels

The Mortara device automatically provides up to 10 textual labels from 236 possibilities for each participant, such as sinus rhythm, normal ECG, and atrial fibrillation. These labels were produced by application of the Minnesota coding system which is a heuristic scheme defined [56]. Sinus rhythm and normal ECG are the most commonly-observed textual labels in our dataset, representing 83.2% and 44.1% of all records, respectively.

The outputs of the Mortara device included a label for abnormality for the waveform (abnormal ECG), which was coded as 1 for abnormality and 0 for normality.

3.4.1 Labels for one-class novelty detection:

Sinus rhythm, normal ECG, and abnormal ECG=0 were the only three labels available in our dataset that were used as potential criteria for defining normality. There were 19,925, 10,550, and
18,397 records labelled sinus rhythm, normal ECG, and abnormal ECG=0, respectively. Sinus rhythm and abnormal ECG=0 had 15,628 records in overlap. Normal ECG, which was a subset of the union of sinus rhythm and abnormal ECG=0, included 10,550 records. There are only 1,675 complete records without any of the three labels for defining normality (Figure 2).

Since the labels were produced according to the heuristic Minnesota criteria, their relationships to diseases require future study. In addition, these criteria have a large number of rules in disagreement due to contradictions and inconsistencies in the coding system. As a result, it may not be convincing to treat any of these labels alone as being the “gold standard” for use in defining normality when training a classification algorithm. This report compares the performance of three algorithms against five reasonable criteria. The five combinations considered by us are (C1) sinus rhythm, (C2) normal ECG, (C3) abnormal ECG=0, (C4) sinus rhythm AND abnormal ECG=0, and (C5) sinus rhythm OR abnormal ECG=0. These will be referred to as the “normal criteria” C1-5.

**Figure 2 Relationship of Sinus Rhythm, Normal ECG, and Abnormal ECG=0.** The number indicate how many records in the continuous area are. For example, the number 4374 means there 4374 records labelled sinus rhythm but not abnormal ECG=0.
3.4.2 Labels for four-class classification:

For 4-class classification of “normal”, “arrhythmia”, “ischemia” and “hypertrophy”, it is important to construct class models as precisely as possible. We also require criteria for the 4-class setting, and here we consider schemes according to [57] which are shown in Table 1.

*Table 1 Four class inclusion criteria and class sizes*

<table>
<thead>
<tr>
<th>Class</th>
<th>Number (%)</th>
<th>Inclusion criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>10803 (43.2)</td>
<td>Normal ECG</td>
</tr>
<tr>
<td>Arrhythmia</td>
<td>2162 (8.6)</td>
<td>Abnormal rhythm, atrial fibrillation, early repolarization, preexcitation, premature ectopic beats, ectopia, blocks, uncertain rhythm</td>
</tr>
<tr>
<td>Ischemia</td>
<td>1868 (7.5)</td>
<td>Explicitly stated “ischemia”</td>
</tr>
<tr>
<td>Hypertrophy</td>
<td>3761 (15.0)</td>
<td>Hypertrophy or enlargement</td>
</tr>
<tr>
<td>Unclassified</td>
<td>6425 (25.7)</td>
<td>None of the above</td>
</tr>
</tbody>
</table>

3.5 Health Insurance Data

The Health Insurance data of 25,001 out of the 25,019 participants were also available, which includes 5 versions of 80 endpoints. The 5 versions each represents a data source: death certificates, underlying cause of death from death certificates, routine disease report data, health insurance event data, and any of the above. The endpoints are disease classes and sub-classes, including vascular disease, malignant neoplasms, disease of the respiratory system, infectious and parasitic disease, diabetes, liver cirrhosis, chronic kidney disease, stroke, disorders of the eye lens, and external causes. These endpoints happened prior to ECG acquisition, for up to 5 years. The time of onset of these causes were also provided, as well as 1,126,619 procedures (ranging from “dimming the light on the ward” to surgery) associated with these causes. Due to the sparsity of the data, cardiovascular diseases from all data sources were used as labels to compare with the
model predictions.

4. PREPROCESSING AND VISUALISATION

4.1 Feature Selection

*Table 2 Mortara features and 4 blood pressure features. Features are excluded either because they can be expressed as functions of other features or because they are constant*

<table>
<thead>
<tr>
<th>Included features</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td></td>
</tr>
<tr>
<td>Average RR interval</td>
<td>Average distance between two R peaks</td>
</tr>
<tr>
<td>QRS offset</td>
<td>End of the QRS complex</td>
</tr>
<tr>
<td>P wave duration</td>
<td>Interval between Q wave onset and T wave offset</td>
</tr>
<tr>
<td>PR interval</td>
<td>Determined by deflects of P wave in different leads</td>
</tr>
<tr>
<td>QRS duration</td>
<td>Interval between Q wave onset and T wave offset</td>
</tr>
<tr>
<td>QT duration</td>
<td>Determined by deflects of QRS wave in different leads</td>
</tr>
<tr>
<td>P axis</td>
<td>Determined by deflects of T wave in different leads</td>
</tr>
<tr>
<td>QRS axis</td>
<td>Systolic blood pressure in the first resurvey</td>
</tr>
<tr>
<td>T axis</td>
<td>Diastolic blood pressure in the first resurvey</td>
</tr>
<tr>
<td>SBP first</td>
<td>Diastolic blood pressure in the first resurvey</td>
</tr>
<tr>
<td>SBP second</td>
<td>Systolic blood pressure in the second resurvey</td>
</tr>
<tr>
<td>DBP first</td>
<td>Diastolic blood pressure in the first resurvey</td>
</tr>
<tr>
<td>DBP second</td>
<td>Diastolic blood pressure in the first resurvey</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Excluded Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ventricular rate</td>
</tr>
<tr>
<td>R peak</td>
</tr>
<tr>
<td>P wave onset</td>
</tr>
<tr>
<td>P wave offset</td>
</tr>
<tr>
<td>QRS onset</td>
</tr>
<tr>
<td>T wave offset</td>
</tr>
<tr>
<td>QT&lt;sub&gt;c&lt;/sub&gt; duration</td>
</tr>
<tr>
<td>QT&lt;sub&gt;cB&lt;/sub&gt; duration</td>
</tr>
<tr>
<td>QT&lt;sub&gt;cF&lt;/sub&gt; duration</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
&= \frac{60000 \text{ Average RR interval}}{\text{always at 500ms}} \\
= Q \text{ offset} - PR \text{ duration} - QRS \text{ duration} \\
= Q \text{ onset} + QT \text{ duration} \\
= \frac{\text{average RR interval}}{QT \text{ duration}} \\
= (\text{average RR interval})^{1/3}
\end{align*}
\]

Of the 19 features available from the Mortara device, 9 were excluded because they can be expressed as functions of other features or because they have constant values. These are shown in
Table 2. We removed records with missing values for any of the features, and data for 24,369 participants were thus retained for use in our study.

4.2 Visualisation

We performed a 2-D visualisation of the 14-dimensional feature space according to the 5 normal criteria, C1-C5, using a Gaussian process latent variable model (GPLVM), which projects high-dimensional data into a lower-dimensional subspace. See appendix B for mathematical details. Here we use GPLVM instead of Principle Component Analysis (PCA) because PCA assumes linear relation of the latent variables and the observed variables, while we prefer to relax this assumption by using a dimension reduction method that can capture nonlinear relations. It may be seen from figures 3 that substantial overlap exists between the “normal” and “abnormal” classes, and among normal, arrhythmia, ischemia, and hypertrophy classes, as is expected for a complex, realistic medical application.

*Figure 3* 2-D visualisation of 500 normal and 500 abnormal data randomly selected according to the criteria C5 (left) and 100 data points randomly selected from each of the four classes (right)
4.3 Signal Quality Evaluation

A Signal Quality Index (SQI) was evaluated for all 12-lead ECG signals independently using in-house software [58]. The SQI $\in [0, 1]$, depends on the agreement of two peak detectors concerning the positions of the R-peaks. Close agreement yields a high SQI, which corresponds to high signal quality.

97.11% of all 12-lead waveforms were deemed to be of high quality (SQI $\geq 0.9$). Each lead has at least 92% data with SQI $\geq 0.9$. Lead V4 has the highest proportion (99.2%) of good-quality signals (SQI $\geq 0.9$). The signal quality of our dataset is therefore deemed to be sufficient to classify at least 92% of the 24,369 participants with complete ECG records.

4.4 Feature Extraction

6 additional features are extracted from the “typical cycles” from each of the 12 leads: amplitudes of P, Q, R, S, and T waves relative to the baseline, and ST level (which was approximated as the level of the ECG observed at the time given by the QRS offset). The baseline was approximated as the average level of the segment between P offset and Q onset, because this segment is used to define ST deviation in the clinic. The positions of the onsets and offsets of these waves are provided by the Mortara device (part of the Mortara features, Figure 4).
5. ANALYSIS

5.1 One-class classification

We present three methods to predict the posterior probability of a feature vector belonging to the abnormal class, and hence predict their cardiovascular disease risk. The entire analysis process is illustrated in Figure 5, and is described in detail as follows:

5.1.1 Cross-validation and partitioning of training and test sets

We performed 5-fold cross-validation by permuting the entire dataset and assigning a different 20% for each fold of cross-validation. Results shown later are the mean values over this 5-fold cross-validation. All sets were normalised column-wise according to the mean and standard deviation of the training set.

5.1.2 Balancing of the test sets

To make a fair comparison between the normal criteria C1-C5 which have different class ratios (i.e. balance between normal and abnormal data), we used the accuracy in balanced sets and AUC.
in both balanced and unbalanced sets for model evaluation.

We therefore created a balanced test set (a subset of the unbalanced test set), containing all abnormal test data and the same number of normal data. The training set remained unbalanced.

Figure 5 Analysis framework for one-class classification
5.1.3 Statistics

The definitions we used for accuracy, true positive rate (TPR), and false positive rate (FPR) are:

\[ \text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \]  
\[ \text{TPR} = \frac{TP}{TP+FN} \]  
\[ \text{FPR} = \frac{TP}{TP+FN} \]

where TP: true positive, TN: true negative, FP: false positive, FN: false negative.

5.1.4 Generative KDE

We adapted the model described in [50]. In brief, the “normal” class probability density function (pdf) was learned from the training set by placing a multivariate Gaussian distribution on each 14-dimensional data point. For ease of computation, we performed k-means clustering to summarise the “normal” data with 500 cluster centres in the 14-dimensional space. Only the most “normal” (i.e., those labelled “normal ECG”) were used in clustering. The data likelihood is calculated via:

\[ p(x) = \frac{1}{N(2\pi)^{D/2}\sigma^D} \sum_{i=1}^{N} e^{-\frac{|x-x_i|^2}{2\sigma^2}} \]  

A novelty score, y, is then calculated using equation 5.

\[ y(x) = - \log p(x) \]

We propose treating this novelty score as a univariate summary of the 14-dimensional data, which may then subsequently used by probabilistic models to predict the probability of test data belonging to the abnormal class. First we learn the likelihood \( p(y|C) \) in the training set, by performing a kernel density estimation of the class-specific pdf. For the unbalanced test set, the
prior P(C) is set to the class ratio of the training set; for the balanced set, the class prior equals 0.5. The posterior is

$$p(C|y) = \frac{p(y|C)p(C)}{p(y)}$$  \hspace{0.5cm} (6)$$

The posterior $P_{test}(C|y)$ is thresholded at 0.5 for classification.

### 5.1.5 Discriminative KDE

Alternatively we can feed the novelty score into a discriminative framework by solving the inference problem $p(C|y)$ directly [59]. The posterior is learned in the training set by binning $y$ and calculating the frequency of observing a particular class in a bin (figure 6). For example, for “normal” class $C_0$, the posterior in each bin is calculated as the proportion of data points belonging to $C_0$ in that bin. Ideally, the bin size should be determined by cross-validation [50]. In this report, the bin size was set to $y = 1$. The posterior of a set with a different prior, in this report the balanced test set, is calculated according to equation 7. Similarly, the posterior is thresholded at 0.5 for classification.

![Figure 6](image)

*Figure 6 Learning the posterior by binning the novelty score $y$. The distribution of $y$ was binned and in each bin the frequency of observing the class $C$ was calculated to estimate the posterior in the training set.*
\[ p_{test}(C|y) = \frac{p_{train}(C|y)p_{train}(y)p_{test}(C)}{p_{train}(C)p_{test}(y)} \] (7)

5.1.6 Discriminative SVM for one-class classification

To compare the results of KDE we also used an SVM. See appendix C for mathematical formulations of SVM. Here we use Gaussian kernel and select the coefficients C and σ by grid search via 5-fold cross-validation on the training set. The classification score is mapped to probabilities and thus the training set posterior \( p(C|y) \) is learned. For an unbalanced test set, the posterior is estimated to be equal to the training set, while the balanced test set posterior is reweighted using equation 7.

5.2 Four-class classification

5.2.1 Constructing the training and test sets

We obtained balanced training and test sets by taking all data from the smallest class (Table 1), and the same number of data points from each of the other classes were randomly selected to construct the training and test sets for 5-fold cross-validation. For example, the four-class balanced training-and-test set contains \( 1868 \times 4 = 7472 \) datapoints. To illustrate the distinctiveness of each class, three-class and two-class classifications of any combinations of the “normal”, “ischemia”, “arrhythmia”, and “hypertrophy” were also performed for comparison using the same approach to balance classes.

5.2.2 Training the 4-class model

A \( P(C|y) \) was estimated for each of the classes using support vector regression (Appendix C) in a one-vs-all approach; i.e. the regressor \( i \) was learned in a training set with only the class \( i \) labelled
1 and other classes were labelled 0. The class probability \( P(C|x) \) was calculated from the predicted value of the regressor \( i \) according to equation 8 (See Appendix C for the derivation of this equation). Finally the data point was classified to the class \( i \) which maximises the probability:

\[
P_i(C_i|x) = \frac{e^{\frac{|1-y_i|}{\sigma_i}}}{2\sqrt{\pi} \sum_{j=1}^{4} P_j(C_j|x)}, \quad i = 1, 2, 3, 4
\]  

(8)

5.2.3 Risk prediction

The 4-class model were used to predict the 24,344 participants with complete features in the health insurance dataset, using cardiovascular endpoints from all data sources for comparison. All participants predicted as arrhythmia, ischemia, or hypertrophy were included in the “Predicted CVD” class.

5.3 Results and Discussion

5.3.1 One-class classification

Under all 5 criteria C1-C5, data from the normal class take overall lower novelty scores than data from the abnormal class, as shown in figure 7. However, the two classes are largely overlapped in the peak area of the abnormal class, suggesting that the generative KDE may not be able to separate them easily. Figure 7 shows the likelihood under the criteria C5, and the results for other criteria were similar.
In order to rule out the influence of different test sets to classification accuracy, we first evaluated the performance of the discriminative SVM and generative KDE with the same (unbalanced) test set. Considering the substantiate overlap shown by the visualisation (Figure 3) and the likelihood (Figure 7), the generative KDE and discriminative SVM have relatively high AUC. Under criterion C5, the AUC is high for both methods. The SVM model achieved higher AUC for all criteria, suggesting it may be a more robust method than the KDE (Figure 8 and Table 3).

**Table 3 AUC under normal criteria C1-C5 in the same test set**

<table>
<thead>
<tr>
<th>Normal Criteria</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.82</td>
<td>0.78</td>
<td>0.79</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>KDE</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
<td>0.75</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Using the balanced set, the discriminative SVM achieved high accuracy, 71.1% to 75.6%, for all 5 criteria, while the generative KDE has a comparable result (74.8%) using criterion C5 (Table 4). The discriminative KDE has similar AUC values as the generative KDE, but lower accuracy, which implies a better optimisation of this method may be needed. Figure 8 shows that the most distinctive normal criterion is different for each method. For the generative KDE and the discriminative SVM the best-performing criterion is C5, and C1, respectively, suggesting different algorithms may favour different criteria.

**Table 4 AUC and Accuracy of predicting the 5 normal criteria by generative KDE, discriminative KDE, and discriminative SVM in the balanced sets.**

<table>
<thead>
<tr>
<th></th>
<th>Discriminative SVM</th>
<th></th>
<th>Generative KDE</th>
<th></th>
<th>Discriminative KDE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Accuracy %</td>
<td>AUC</td>
<td>Accuracy %</td>
<td>AUC</td>
<td>Accuracy %</td>
</tr>
<tr>
<td>C1</td>
<td>0.79</td>
<td>71.7</td>
<td>0.73</td>
<td>67.2</td>
<td>0.73</td>
<td>59.6</td>
</tr>
<tr>
<td>C2</td>
<td>0.79</td>
<td>71.6</td>
<td>0.73</td>
<td>63.5</td>
<td>0.73</td>
<td>61.1</td>
</tr>
<tr>
<td>C3</td>
<td>0.82</td>
<td>77.1</td>
<td>0.72</td>
<td>65.1</td>
<td>0.72</td>
<td>59.3</td>
</tr>
<tr>
<td>C4</td>
<td>0.80</td>
<td>73.7</td>
<td>0.75</td>
<td>64.5</td>
<td>0.75</td>
<td>61.4</td>
</tr>
<tr>
<td>C5</td>
<td>0.83</td>
<td>75.6</td>
<td>0.81</td>
<td>74.8</td>
<td>0.80</td>
<td>60.6</td>
</tr>
</tbody>
</table>

**Figure 8 ROC curves of the same unbalanced test set under the five normal criteria**
It is unexpected that the most stringent criterion, C3, being the subset of all other 4 criteria, has not yielded best results with any of the algorithms considered. The criterion C5 is predicted accurately by both the generative KDE and the discriminative SVM, suggesting that it may be more appropriate for use as the “gold standard” for training algorithms for one-class classification.

5.3.2 Four-class classification

The four-class classification results using support vector regression are shown in Figure 9. Accuracies between the two classes on the nodes; numbers in the centres of the triangle are classification accuracies of the 3 classes on the nodes of the triangle. The red and black numbers are results with and without the 72 features derived from the ECG, over and above the basic set of the 10 features provided by the Moratra device and the 4 blood pressure features, respectively.

The 72 new features improved the results in all cases, most markedly in classifications involving ischemia and hypertrophy. This agrees with our expectation since the 10 Mortara features do not contain information concerning the amplitudes of the peaks, while ischemia and hypertrophy were highly correlated with amplitude abnormalities, especially ST-levels, and R-S amplitudes.

It is encouraging that the classification accuracies with the new features are 30 to 50 percentage points higher than those that would be obtained by chance, suggesting machine learning methods can achieve high agreement with clinical knowledge, without resorting to a complex rule-based system.

In comparison with the HI dataset endpoints, 8,597 participants who were labelled CVD negative in the HI dataset were predicted with CVD, while 1607 participants labelled CVD positive
were predicted as being “normal”. The substantial time difference between ECG acquisition and the endpoint onsets (as well as the short duration of the ECG recordings) are major contributors to this discrepancy. It is possible that participants may have developed or recovered from CVD during the time elapsed between the label in the health insurance dataset and the acquisition of ECG data.

![Diagram showing classification accuracies](image)

**Figure 9 Classification accuracies (see the text for description)**

**Table 5 Comparison of predicted CVD with health insurance records**

<table>
<thead>
<tr>
<th></th>
<th>Predicted CVD</th>
<th>Predicted Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVD positive</td>
<td>1882</td>
<td>1607</td>
</tr>
<tr>
<td>CVD negative</td>
<td>8597</td>
<td>12258</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS AND FUTURE WORK

6.1 Summary of work to date

We have addressed the task of risk evaluation using ECG of about 25,000 participants by (i) building three novelty detection (one-class classification) models to detect abnormality with a generative KDE, a discriminative KDE, and a discriminative SVM, and (ii) a 4-class model for
probabilistic risk prediction of normal, arrhythmia, ischemia, and hypertrophy using 82 features
extracted from ECG times series and 4 blood pressure data from about 25,000 participants, using
support vector regression. We tested the validity of our one-class and four-class models with
Mortara labels (i.e., those from a complex, heuristic clinical algorithm). Subsequently we used our
four-class model to predict four-class probabilistic risk scores on the same cohort, and then
compared our results with their disease history in the health insurance dataset.

Specially, to address the second research question proposed at the end of section 2, we have
proposed a framework that reweights the posterior trained in a highly unbalanced training set. This
method is especially desirable in novelty detection, where balancing the training set means loss of
the majority of the data.

The first task of this research was to detect “abnormality”, by exploring different one-class
novelty detection algorithms under various criteria of “normality”. The algorithms favoured
criteria C5, which was the least stringent of all possibilities. In view of the relatively good
performance of the discriminative SVM, in the second task of this study we modelled the “normal”,
“arrhythmia”, “ischemia”, and “hypertrophy” using multiclass support vector regression, aiming
to produce accurate models for prediction of the unclassified data points according to the labels.
The encouraging results suggest the multiclass models may be appropriate to predict the
probability of class membership of the “borderline” data that are otherwise difficult to classify.
We can further improve the classification accuracy by extracting more features, such as heart-rate
variability and T-wave alternans.

Our novelty detection task is not a typical supervised machine learning problem, due to the
absence of the “gold” standard. We proposed systematical evaluation of the model against all
“silver” standards, and base our evaluation on the assumption that if our predictions are sufficiently close to the silver standards, they are also sufficiently close to the latent “gold” standard.

The $P_i(C_i|x)$ in the four-class model (eq. 8) used in classification is actually a “pseudo-posterior”, because it was calculated from the probability of regression scores, whose pdf was calculated in a parametric approach (via the Laplace distribution). The regression score itself does not have a probabilistic interpretation, and we can make the model more mathematically sound by calculating the “true posterior” according to eq. 6 and predict four probabilistic risk scores for each participants. Also, the four classes (normal, ischemia, arrhythmia, and hypertrophy) are treated as independent so far. With a “true posterior” we can produce conditional risk scores based on any relationships that may exist between the four classes.

The original 10s signal may lend more information than the “typical cycle” as the former contains more time-dependent information than the latter. The length of the signal is a major limitation to our feature extraction, because many informative features such as ST-level need longer (> 60s) signals to be evaluated accurately [60]. Features may be extracted from the 10s signal using recurrent neural networks (RNN) and compared with clinical indices such as the Sokolow Index [61]. Future work will link our analysis of ECG data to other epidemiology data such as body mass index (BMI), blood pressure, and stroke risk scores available via the China Kadoorie Biobank.

6.2 Immediate future work

6.2.1 Further analysis

Our immediate future work will include (i) performing additional analyses with existing models to better compare with the literature, reporting on metrics that would permit direct comparison; (ii)
producing Bayesian probabilistic risk scores to introduce “rejection options” (e.g., “cannot classify”) and conditional risk scores; and (iii) comparing our features with the clinical-standard Sokolow index [62] to evaluate the improvement to this well-understood but simplistic existing method, and iv) conceptualise our framework of evaluating against multiple “silver” standards with the absence of the “gold” standard using latent variable model. We denote the latent “gold” standard $Z = \{z_n\}$, and the noisy observations of $Z$, i.e. the silver standards as $T = \{t_{nk}\}$, with the superscript indexing the “silver” standards and subscript indexing the participant, and model $P(T|Z)$ using appropriate parametric (for example multinomial) or nonparametric models, and study the joint distribution $P(X,T)$, where $X$ is data matrix.

We will also perform a retrospective analysis taking advantage of the occurrence time of the diseases in the health insurance data, to illustrate how our prediction improves as we exclude distant events from ECG acquisition, and further validate our models in the time domain. These tasks will likely be completed by December 2016, when we will discuss with our clinical collaborators about ways to communicate the results of our research to a clinical audience for publication in the epidemiological literature.

### 6.2.2 Epidemiology analysis

We will correlate our results with blood pressure, and subsequently after data acquisition, with body mass index (BMI) and stroke risk scores, to evaluate the contribution of our risk scores to risk stratification. We will then build robust electronic health record (EHR) models with heterogeneous data in the CKB database. The latter will involve consideration of probabilistic models that permit the fusion of categorical data, and which may include sparse techniques for handling the largely-incomplete records in the dataset. These studies will likely be conducted
during Michaelmas Term 2016/2017.

### 6.2.3 “Omics” data analysis

We will then analyze the genomics and proteomics data in the CKB database. These will be continuous or categorical data and can be explored by various machine learning models, including Bayesian non-parametric “Indian buffet processes” for matrices of sparse binary data. Comparison will be performed with appropriate kernel methods, such as sparse logistic kernel machines. We will incorporate our ECG risk scores with -omics data to create comprehensive risk scores, potentially for more disease types than we have initially considered, such as cancer and stroke. These will likely be conducted from Michaelmas to Hilary Term of 2016/2017.

### 6.2.4 Deep learning models for ECG analysis

Cardiac activity is a dynamic system which can be accurately modelled by recurrent neural network (RNN), which differs from feedforward neural network in that there are loops in directed graphs (Figure 10, courtesy [63]). Funahashi et al. proved that an arbitrary dynamic system can be modelled accurately by a continuous recurrent neural network (RNN) [64]. However, the main difficulty in training the RNN is that the conventional backpropagation algorithm requires acyclic structure, thus cannot be applied directly to RNN. Many algorithms have been proposed for training of RNN, including BPTT [65, 66], real time recurrent learning (RTRL) [67], etc. However, they suffer from ‘blowing-up’ or ‘vanishing’ back error derivatives, thus conventional RNN doesn’t perform well when the time elapse required in the short-term memory is relatively long. This problem was studied intensively in [68]. Mathematically, Hochreiter et al. gave the local scale factor between error gradients of arbitrary unit $u$ at time point $t$ back prop to unit $v$ at time $t - q$ is:
\[
\frac{\partial \sigma_v(t-q)}{\partial \sigma_u(t)} = \sum_{l_1=1}^{n} \cdots \sum_{l_{q-1}=1}^{n} \prod_{m=1}^{q} f_{lm}'(\text{net}_{lm}(t-m))w_{lm}l_{m-1}
\]  

where

\[
\sigma_k(t) = \begin{cases} 
  f_k'(\text{net}_k(t))(d_k(t) - y^i(t)), & \text{k is output unit} \\
  f_k'(\text{net}_k(t)) \sum_i w_{ik} \sigma_k(t+1), & \text{k is nonoutput unit}
\end{cases}
\]  

is the error of arbitrary unit \( k \) at time point \( t \), where \( d_k(t) \) is the target at time \( t \) for unit \( k \), if \( k \) is an output unit, and

\[
y^i(t) = f_i(\text{net}_i(t))
\]  

is the activation of a non-input unit \( i \) with activation function \( f_i \), and

\[
\text{net}_i(t) = \sum_j w_{ij}y^j(t-1).
\]

If

\[
|f_{lm}'(\text{net}_{lm}(t-m))w_{lm}l_{m-1}| > 1.0
\]

the error gradient will explode by the end of backpropagation;

If

\[
|f_{lm}'(\text{net}_{lm}(t-m))w_{lm}l_{m-1}| < 1.0
\]

the error gradient will vanish.

Thus to ensure the parameters can be properly learned the gradient must satisfy

\[
|f_{lm}'(\text{net}_{lm}(t-m))w_{lm}l_{m-1}| = 1.0
\]

In other words, a simple recurrent neural network (SRN) has only a simple recurrent unit in its hidden layers which is usually a sigmoidal function (figure 10 left), and because of the product of multiple sigmoid functions will flatten, an effective RNN structure must contain addition operations. Hochreiter et al. [69] proposed long short-term memory (LSTM) which features gated cells instead of the simple recurring unites in SRN. One of the most commonly used LSTM structure nowadays is the so-called ‘Vanilla network’ proposed by Graves and
Schmidhuber [70]. We follow the presentation in [63] and present the illustration (courtesy [63]) and mathematical formulation as follows:

![Diagram of SRN and LSTM units](image)

**Figure 10** Detailed schematic of the Simple Recurrent Network (SRN) unit (left) and a Long Short-Term Memory block (right) as used in the hidden layers of a recurrent neural network.

in which

\[ z^t = g(W_z x^t + R_z y^{t-1} + b_z) \]  
\[ i^t = \sigma(W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i) \]  
\[ f^t = \sigma(W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f) \]  
\[ c^t = i^t \odot z^t + f^t \odot c^{t-1} \]  
\[ o^t = \sigma(W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o) \]  
\[ y^t = o^t \odot h(c^t) \]

where \( W \) denote the weight matrix for input \( x \), \( R \) is the square matrix linking recurrent units, \( b \) is the bias vector, \( p \) is the peephole vector which can be used to make variations on the structure, with the simplest structure being a unit vector, and functions \( \sigma, g \) and \( h \) are point-wise non-linear activation functions: logistic sigmoid is used for as activation function of the gates \( i, o, f \), and
hyperbolic tangent is usually used as the block input $z$ and output activation function $y$.

Central to the LSTM structure is the addition operation which forces constant error carousel (CEC, equation 15). The input and block output from other memory cells in the hidden layers are not only fed into the cell state, but also to each of the three gates. In learning phase all $W$, $R$, and $b$ are to be learned, which is analogous to training each memory cell to forget, input, or output when appropriate. The LSTM net was shown to be able to effectively retain memory of 1000 time steps [69]. The classic applications of LSTM are on sequential data in which previous data are informative to the latter data, such as in natural language processing and polymorphic music modelling, and LSTM is arguable the most effective and efficient algorithm for predicting sequential data [71]. We refer to [72] for PhD thesis on LSTM, [73] for PhD thesis on training RNN, [74] for applying LSTM on sequential data, [75] for application on image generation, and [76] for gated feedback RNN. Helpful tutorials can be found at http://deeplearning4j.org/lstm and http://people.idsia.ch/~juergen/rnn.html, and open-source software packages can be found at http://people.idsia.ch/~juergen/rnn.html.

We conclude that the LSTM is appropriate for modelling time-series medical data, yet there has not been such a study reported in the literature. One way to model ECG, for example, is to use a subset of “normal” ECG signal as a training set, in which the ECG signal is the input $x$, prediction of time point $t$ later than $x$ being $y(t)$; this allows us to quantify the divergence of prediction $y(t)$ from the original target with some loss function, for example the squared loss. If the error is sufficiently small, we can subsequently use the training model to quantify ‘abnormal’ or ‘borderline’ ECG signals with the same loss function, thus answering the research question: ‘How to quantify normality in time-series signal?’
Alternatively we can monitor the cell state $c(t)$ which may contain interesting hidden patterns in the data, that is, what the model “remembers” in the history of the signal, which may or may not have a ready clinical interpretation.

LSTM network can also analyze 12-leads simultaneously, by setting $\mathbf{x}(t) = (l_1(t), l_2(t), ..., l_{12}(t))^T$, where $l_i(t)$ is the signal value of lead $i$ at time $t$. Thus we may discover cross-lead features in a principled way, parallel to clinical cross-lead indices such as the Sokolow index [62].

In addition, we can adapt the LSTM model for unsupervised learning, by borrowing ideas from the feedforward deep encoder [77], which uses entropy as the loss function thus eliminating the need for the noisy class labels.

If LSTM does not prove effective in the first instance, we will perform a model diagnosis (see section 6.4) and adjust the model or dataset accordingly. In rare cases when the model failure cannot be explained, we can resort to alternative time-series analysis approaches such as Gaussian process (Appendix A) or more methods summarized in [60].

6.3 Plan until completion

As illustrated in the Gantt chart (figure 11), there will likely be biomedical conferences in the summers of 2017 and 2018, and writing and attending for these conferences are planned. A 6-week study abroad program of the CDT Healthcare Innovation program may take place in the summer of 2017 at a research group that would benefit my understanding of the methodology (including collaborating machine learning labs in the USA). From the summer of 2017 focus will be on writing for confirmation of status, which is due on 1 September, 2017. The writing and preparation for this will likely take less than 3 months. The following 6 months will be spent on writing-up for thesis. The final submission deadline is 1 September 2018.
6.4 Risk assessment

There is very low foreseeable risk associated with our research, in terms of data acquisition and implementation of the techniques. All data are stored in the China Kadoorie Biobank (CKB) database, and acquisition and preparation of data have been made a routine which is expected to be completed in about two weeks. When implementing the technique, we use the bias-variance analysis to make decisions on which techniques to evaluate. Given the test set error is high, if the gap between training and test set errors in the learning plot (figure 12) is large, the model has high variances (overfitted); if the gap is small, the model has high bias (underfitted). Approaches for solving underfitting include:

- Decrease regularization, or alter the hyperparameters in the Bayesian priors
correspondingly

- Decrease training size
- Adding more features
- Increase nodes or layers in the neural network

If the model is overfitted, we

- Increase regularization, or alter the hyperparameters in the Bayesian prior correspondingly
- Increase training size
- Decrease number of features (wrapper or filter feature selection)
- Decrease nodes or layers in the neural network

Obtaining better set of features and learning the hyperparameters by evidence approximation will help in both cases. These frameworks will ensure the model is trained in the correct direction.

![Learning curve](image)

*Figure 12 Learning curve for high bias (left) and high variance (right)*

Enough time margin is reserved for any contingencies that might arise while writing-up, and
writing will start substantially ahead of due dates. I have sufficient funding for my DPhil until 1 September 2019, which should incorporate the research and writing-up of my doctoral programme.

Appendix

Appendix A: Gaussian Process Regression

A Gaussian Process (GP) is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution [54]. Intuitively, a Gaussian process is a ‘Gaussian distribution’ (or more precisely a stochastic process) of functions, specified by its covariance function (also termed kernel function), analogous to the Gaussian destruction for scalar or vector variables, which are specified by the mean and (co)variance. The mean of a Gaussian process is often taken as zero to simplify notation, without loss of generality. Therefore, the ‘learning’ phase of Gaussian process regression is to infer the structures and parameters of the covariance functions.

Inference for Gaussian Process regression

GP is motivated by the potential to use the ‘kernel trick’, as illustrated in the ‘weight-space’ view of GP: let us define phi(x) as a mapping from the original data space of D dimension to N dimensional feature space, and let us denote the number of datapoints as n. We model the target function as

\[ f(x) = \phi(x)^T w \]  

(A.1)

with w a N dimensional vector.

The likelihood of the dataset can be written down as

\[ p(y|X, w) = \prod_{i=1}^{n} p(y_i | x_i, w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i}} \exp \left( -\frac{(y_i - x_i^T w)^2}{2\sigma_i^2} \right) \]
\[
\frac{1}{(\sqrt{2\pi\sigma_n^2})^n} \exp \left(-\frac{|y-x^Tw|^2}{2\sigma_n^2}\right) = \mathcal{N}(X^Tw,\sigma_n^2I) \tag{A.2}
\]

substitute \(x\) by \(\phi(x)\). According to the Bayesian equation, the posterior

\[
p(w|y,X) = \frac{p(y|\Phi,w)p(\phi(x))}{p(y|X)} \tag{A.3}
\]

Where \(\Phi\) is the matrix with column vectors \(\phi(x)\). If we put a zero mean Gaussian prior for \(w\)

\[
w \sim \mathcal{N}(0,\Sigma_p) \tag{A.4}
\]

We can write down the posterior by ‘completing the square’ [59]

\[
p(w|X,y) \propto \exp\left(-\frac{1}{2\sigma_n^2}(y - \Phi^Tw)^T(y - \Phi^Tw)\right)\exp\left(-\frac{1}{2}w^T\Sigma_p^{-1}w\right)
\]

\[
\propto \exp\left(-\frac{1}{2}(w - \bar{w})^T\left(\frac{1}{\sigma_n^2}\Phi\Phi^T + \Sigma_p^{-1}\right)(w - \bar{w})\right) \tag{A.5}
\]

In Bayesian regression the parameter \(w\) is marginalized out and give the predictive distribution for the new data \(x_\ast\)

\[
p(f_\ast|x_\ast,X,y) = \int p(f_\ast|\phi(x_\ast),w)p(w|X,y)dw = \int \phi(x_\ast)^Twp(w|X,y)dw
\]

\[
= \mathcal{N}\left(\frac{1}{\sigma_n^2}\phi(x_\ast)^TA^{-1}Xy,\phi(x_\ast)^TA^{-1}\phi(x_\ast)\right) \tag{A.6}
\]

where

\[
A = \sigma_n^{-2}XX^T + E_p^{-1} \tag{A.7}
\]

where we can see to make prediction we need to invert matrix \(A\) which is \(N\) by \(N\), which can be inconvenient if \(N\) is large; alternatively, we can write down equation A.6 in its equivalent form

\[
f_\ast|x_\ast,X,y \sim \mathcal{N}(\phi_\ast^T\Sigma_p\Phi(K + \sigma_n^2I)^{-1}y, \phi_\ast^T\Sigma_p\phi_\ast - \phi_\ast^T\Sigma_p\Phi(K + \sigma_n^2I)^{-1}\Phi^T\Sigma_p\phi_\ast) \tag{A.8}
\]

where

\[
K = \Phi^T\Sigma_p\Phi \tag{A.9}
\]

where we need to invert \((K + \sigma_n^2I)\) which is \(n\) by \(n\), which will be more convenient if \(n<N\). In this formulation the information in \(x\) always comes in the form of inner product of feature vectors,
and since $\Sigma_p$ is positive definite, we can define and define $(\Sigma_p^{1/2})^T \Sigma_p^{1/2} \Sigma_p$, and $k(x, x') = (\phi(x)(\Sigma_p^{1/2}))^T \phi(x)(\Sigma_p^{1/2}) = \psi(x)^T \psi(x)$, which is called covariance function or kernel. Thus we can substitute these inner products with kernels directly, without explicitly compute the basis functions, which is called kernel trick.

An alternative view of GP is the ‘function space’ view, which can simpler notations, and we give definition of GP in this view.

**Definition 1.1 Gaussian Process [54]:**

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution.

The mean of a stochastic process of a function $f(x)$ is defined as

$$m(x) = \mathbb{E}[f(x)] \quad (A.10)$$

Covariance function is defined as

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \quad (A.11)$$

And a function follows Gaussian process is denoted

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad (A.12)$$

Intuitively, a GP is to functions what Gaussian distributon is to variables. To make inference, we use the definition that the joint distribution of the training set $f$ and test set $f_*$ is Gaussian, and without loss of generality we take the mean to be zero, thus we can write down the joint distribution

$$\begin{bmatrix} f \\ f_* \end{bmatrix} = \mathcal{N} \left( 0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right) \quad (A.13)$$

And the predictive distribution can be written down via the ‘split’ of the Gaussian [59]

$$f_*|X, X, f \sim \mathcal{N}(K(X_*, X)K(X, X)^{-1}f, K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*) ) \quad (A.14)$$
For noisy observations, if we assume noise to be mutually independent, we simply add a diagonal term in the training covariance,

\[
\text{cov}(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \tag{A.15}
\]
or

\[
\text{cov}(y) = K(X, X) + \sigma_n^2 I \tag{A.16}
\]

so the joint distribution

\[
\begin{pmatrix}
     y \\
     f_*
\end{pmatrix}
\sim \mathcal{N} \left( 0, \begin{bmatrix}
     K(X, X) + \sigma_n^2 I & K(X, X_*) \\
     K(X_*, X) & K(X_*, X_*)
\end{bmatrix} \right) \tag{A.17}
\]

and predictive distribution, which are the key equations of GP regression.

\[
f_*|X, y, X_* \sim \mathcal{N}(\overline{f_*}, \text{cov}(f_*)) \tag{A.18}
\]

\[
\overline{f_*} \triangleq \mathbb{E}[f_*|X, y, X_*) = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}y \tag{A.19}
\]

\[
\text{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_*) \tag{A.20}
\]

**Inference for Gaussian Process classification**

There are two general approaches to the task of classification, typically described as generative and discriminative. The generative approach models the class-conditional distribution \(p(x|C)\), then calculates the posterior \(p(C|x)\) according to Bayes’ equation (equation 6). The discriminative approach directly models the posterior \(p(C|x)\). Gaussian process classification is a discriminative method.

The output of GP regression is a stochastic process which does not naturally lie between [0,1]. The general idea of using GP for classification is to ‘squash’ the output of GP to lie between [0,1] with a link function (also termed activation function). The link function is usually chosen to be logistic regression, which rises naturally if we write posterior as
\[ p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1)+p(\mathbf{x}|C_2)p(C_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \]  

where

\[ a = \log \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} \]  

\[ \sigma(a) = \frac{1}{1+\exp(-a)} \]

The multiclass equivalent is the softmax function, which can be derived similarly. For all exponential families, the posterior is a logistic function of a linear combination of the input variables or basis functions.

Suppose we put a GP prior on a latent function, then squash it through a logistic function to obtain a prior. The inference includes two steps, first to calculate the predictive distribution of the latent function given a new test set:

\[ p(f_*|X, \mathbf{y}, \mathbf{x}_*) = \int p(f_*|X, \mathbf{x}_*, f)p(f|X, \mathbf{y})df \]  

then squash it into the predictive probability

\[ \overline{\pi_*} \triangleq p(y_* = +1|X, \mathbf{y}, \mathbf{x}_*) = \int \sigma(f_*)p(f_*|X, \mathbf{y}, \mathbf{x}_*)df_* \]  

The non-Gaussian likelihood in (A.24) makes the integral analytically intractable, thus we need to either evaluate it by analytical approximation (for example Laplace approximation[78] or expectation propagation (EP)[79] or numerical solutions based on Monte Carlo. The optimisation details will not be covered in this report.

**Appendix B: The Gaussian Process Latent Variable Model**

The Gaussian Process Latent Variable Model (GPLVM) was proposed by Lawrence et al. [80]. It is a natural extension of the linearity of probabilistic principle component analysis (PPCA) using a nonlinear kernel. Here we denote \( Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\} \) as observed high-dimensional variables,
and \( X = \{x_1, x_2, \ldots, x_N \} \) as the low-dimensional variables. PPCA assumes both \( X \) and \( Y \) are Gaussian distributed, and \( Y \) is linearly related to \( X \). Without loss of generality, we use a zero-mean, univariate Gaussian prior for \( X \):

\[
p(x_n) = \mathcal{N}(x_n|0, I)
\]

We can write the conditional distribution as

\[
p(y_n|x_n, W, \beta) = \mathcal{N}(y_n|Wx_n, \beta^{-1}I)
\]

where \( \beta \) is the precision. There are two ways to marginalise the conditional distribution and maximize the resulting marginalised likelihood, which will lead to equivalent formulations of PPCA. The first way is to marginalized over \( X \) and maximize with respect to \( W \) [81, 82] and the other way is to marginalise over \( W \) and marginalise over \( X \) [80]. A Gaussian Process will naturally appear as a result of the latter approach, as shown in more detail here:

Assume \( Y \) is independently identically distributed (i.i.d.), and give a prior of

\[
p(W) = \prod_{i=1}^{D} \mathcal{N}(w_i|0, \alpha^{-1}I)
\]

where \( w_i \) is the \( i \)th row of \( W \), we can write down the marginalised likelihood:

\[
p(Y|X, \beta) = \frac{1}{(2\pi)^{DN}D} \exp\left(-\frac{1}{2} tr(K^{-1}YY^T)\right)
\]

and where \( D \) is the original (higher) dimension, and \( N \) is the number of data in \( Y \),

\[
K = \alpha XX^T + \beta^{-1}I
\]

\[
X = [x_1, \ldots, x_N]^T
\]

\[
L = -\frac{DN}{2} \log(2\pi) - \frac{D}{2} \log|K| - \frac{1}{2} tr(K^{-1}YY^T)
\]

To maximize \( L \) we solve for the zero point of the gradient:

\[
\frac{\partial L}{\partial X} = \alpha K^{-1}YY^T K^{-1}X - \alpha DK^{-1}X = 0
\]

\[
X = U_q L V^T
\]
where $q$ is the number of principle components of interest, $U_q$ is the $N$ by $q$ matrix with column vectors the eigenvectors of $YY^T$. $V^T$ is an arbitrary $q$ by $q$ orthogonal matrix. $L$ is the $q$ by $q$ diagonal matrix with elements:

$$l_j = \left( \frac{\lambda_j}{\alpha D} - \frac{1}{\beta \alpha} \right)^{-\frac{1}{2}} \tag{B.10}$$

which is equivalent to PPCA, and take limit $\alpha, \beta \to \infty$, we recover standard PCA.

We recognize that (B.7) is a product of D independent GPs, whose linear covariance function is given by

$$K = \alpha XX^T + \beta^{-1}I \tag{B.11}$$

A natural extension is to substitute $K$ with a nonlinear kernel, such as a squared exponential kernel:

$$k_{n,m} = \alpha \exp \left( -\frac{\gamma}{2} (x_n - x_m)^T(x_n - x_m) \right) + \delta_{nm} \beta^{-1} \tag{B.12}$$

Note that due to the nonlinearity of $K$, $L$ is no longer convex, therefore we need numeric optimisation algorithm such as scaled conjugate gradients (SCG) [83], and due to the presence of local minima the solution is not unique. Note that the solution of PCA is subject to arbitrary rotation, represented by $R$, thus is not unique either. Optimisation details is beyond the scope of this report.

**Appendix C: Support Vector Machines**

The support vector machine (SVM) was proposed by Vapnik et al. [84]. Let’s consider two-class first. Suppose the two classes are linearly separable in feature space, that is, the $N$ dimensional features can be separate by a $N-1$ dimensional hyperplane. And we denote this hyperplane

$$y(x) = w^T \phi(x) + b \tag{C.1}$$

where $\phi(x)$ is the basis functions that map the data from the original $D$ dimension to the $N$
dimensional feature space. The hyperplane is called the ‘decision boundary’.

The SVM is a large margin classifier, which means it maximizes the distance between the decision boundary and the margin. If we denote the class targets as \( t_n \in \{-1, 1\} \), then SVM requires all data satisfy

\[
 t_n (w^T \phi(x_n) + b) \geq 1, \quad n = 1, \ldots, N. \tag{C.2}
\]

which is easily interpreted as all data should lie within the correct boundary, and the number on the right-hand side of the inequality can be any arbitrary positive number, because the distance is \( t_n y(x_n) / \|w\| \), which can be maximized by solving

\[
 \arg \min_{w, b} \frac{1}{2} \|w\|^2
\]

For linearly inseparable data, the SVM relaxes the boundary condition (C.2) by introducing a non-negative slack variable \( \xi_n \), and the optimization objective is (C.4) subject to (C.5). The parameter \( C \) controls the trade-off between fitting the data exactly and allowing some data to be misclassified, to improve generality.

\[
 C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|w\|
\]

\[
 t_n (w^T \phi(x_n) + b) \geq 1 - \xi_n, \quad n = 1, \ldots, N. \tag{C.5}
\]

To implement the optimizations we first introduce a Lagrange multiplier, giving the Lagrangian equations (C.6) for linearly separable data. If we make use of equations C.7 and C.8, we can rearrange the Lagrange equation into the dual representation (C.9), which shows the kernel explicitly, enabling us to apply ‘kernel trick’, by defining the kernel function \( k(x_n, x_m) = \)
\(\phi(x)^T \phi(x')\) directly, which is often more efficient than calculating the basis function \(\phi(x)\) explicitly. The optimization problem for linearly separable case thus becomes minimizing C.9 with respect to \(a\) and subject to C.10 and C.11.

\[
L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{n=1}^N a_n [t_n (w^T \phi(x_n + b)) - 1]
\]

(C.6)

\[
w = \sum_{n=1}^N a_n t_n \phi(x_n)
\]

(C.7)

\[
0 = \sum_{n=1}^N a_n t_n
\]

(C.8)

\[
\bar{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m)
\]

(C.9)

\[
a_n \geq 0, n = 1, ..., N,
\]

(C.10)

\[
\sum_{n=1}^N a_n t_n = 0.
\]

(C.11)

And for linearly inseparable classes is also (C.9), with the boundary condition:

\[
0 \leq a_n \leq C
\]

(C.12)

\[
\sum_{n=1}^N a_n t_n = 0.
\]

(C.13)

This formulation is referred to as C-SVM in literature. An equivalent formulation was proposed by Schölkopf [85]: optimize \(\bar{L}(a)\)

\[
\bar{L}(a) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m)
\]

(C.14)

subject to

\[
0 \leq a_n \leq \frac{1}{N}
\]

(C.15)
\[
\sum_{n=1}^{N} a_n t_n = 0. \quad \text{(C.16)}
\]
\[
\sum_{n=1}^{N} a_n \geq \nu. \quad \text{(C.17)}
\]

Which is referred to as \( \nu - SVM \), and the parameter \( \nu \) has the interpretation as the upperbound of the fraction of data points lie on the wrong side of the margin, but not necessarily misclassified.

The parameters of SVM are \( C \) or \( \nu \), and the parameters of the kernel function.

For a K-class SVM (K>2), two common strategies are one-vs-one and one-vs-rest. The former trains a two-class classifier for every pair of classes, thus K(K-1)/2 classifiers in total, then classifies the data to the class with the smallest loss \( L(w, b, a) \). However the one-vs-one classifiers do not have readily comparable \( L(w, b, a) \). The latter trains K two-class classifiers, for each classifier only one class is the positive class. Although both approaches will have ambiguous regions, as shown in [59], the one-vs-rest approach is still the most commonly used multiclass SVM method [54].

Support vector machine can also perform regression. One of the standard formulations is the \( \epsilon \) - insensitive support vector regression (SVR). The error function is

\[
E_\epsilon(y(x) - t) = \begin{cases} 0, & \text{if } |y(x) - t| < \epsilon \\ |y(x) - t| - \epsilon, & \text{otherwise} \end{cases} \quad \text{(C.18)}
\]

which penalizes only the data lying outside of the ‘\( \epsilon \)-tube’. The minimization objective is

\[
C \sum_{n=1}^{N} E_\epsilon(y(x) - t) + \frac{1}{2} ||w||^2 \quad \text{(C.19)}
\]

Subject to

\[
t_n \leq y(x_n) + \epsilon + \xi_n \quad \text{(C.20)}
\]
\[
t_n \geq y(x_n) + \epsilon + \xi_n \quad \text{(C.21)}
\]
For probabilistic SVR we follow [86] to model the conditional distribution $p(y|t)$ as a Laplace distribution with mean $t \in \{0,1\}$, the merit of which is studies in [87].

$$p(y|t) = p(z) = \frac{1}{2\sigma} e^{-\frac{|z|}{\sigma}}$$  \hspace{1cm} (C.22)

where $z = y - t$, and $\sigma$ was obtained during training through maximum likelihood. We train the four-class via one-vs-rest approach, and normalize the resulting conditional distribution, giving the posteriors (Equation 8).

**Bibliography**

Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society
Endorsed by the International Society for Computerized Electrocardiology. Journal of the

62. Hancock, E.W., et al., AHA/ACCF/HRS Recommendations for the Standardization and
Interpretation of the Electrocardiogram: Part V: Electrocardiogram Changes Associated
With Cardiac Chamber Hypertrophy A Scientific Statement From the American Heart
Association Electrocardiography and Arrhythmias Committee, Council on Clinical
Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society
Endorsed by the International Society for Computerized Electrocardiology. Journal of the


64. Funahashi, K.-i. and Y. Nakamura, Approximation of dynamical systems by continuous

65. Williams, R.J. and D. Zipser, Gradient-based learning algorithms for recurrent networks
and their computational complexity. Back-propagation: Theory, architectures and

66. Werbos, P.J., Generalization of backpropagation with application to a recurrent gas

University of Cambridge Department of Engineering.

68. Bengio, Y., P. Simard, and P. Frasconi, Learning long-term dependencies with gradient


data with recurrent neural networks. in Proceedings of the 23rd international conference
on Machine learning. 2006. ACM.


Hannover.


77. Hinton, G.E. and R.R. Salakhutdinov, Reducing the dimensionality of data with neural

78. Williams, C.K. and D. Barber, Bayesian classification with Gaussian processes. IEEE