Novel algorithms and rating methods for high-performance ECG classification

Der Naturwissenschaftlichen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur

Erlangung des Doktorgrades Dr. rer. nat.

vorgelegt von

Christian Rockstroh
aus Sonneberg
Als Dissertation genehmigt
von der Naturwissenschaftlichen Fakultät
der Friedrich-Alexander Universität Erlangen-Nürnberg
Tag der mündlichen Prüfung: 21.11.2013

Vorsitzender des Promotionsorgans: Prof. Dr. Johannes Barth

Gutachter: Prof. Dr. Bernhard Hensel

Prof. Dr. Björn Eskofier
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>Kurzfassung</td>
<td>1</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>2 ECG-Basics</strong></td>
<td>9</td>
</tr>
<tr>
<td>2.1 The origin of the ECG</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Data material</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Mathematical framework</td>
<td>17</td>
</tr>
<tr>
<td>2.4 The ρ-Score: a framework for comparing ECG classifiers</td>
<td>23</td>
</tr>
<tr>
<td><strong>3 Preprocessing</strong></td>
<td>31</td>
</tr>
<tr>
<td>3.1 FIR-Filter</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Filtering noise and artifacts</td>
<td>34</td>
</tr>
<tr>
<td>3.3 QRS-complex detection and delineation</td>
<td>37</td>
</tr>
<tr>
<td>3.4 ECG-segmentation and normalization</td>
<td>41</td>
</tr>
<tr>
<td><strong>4 Feature Extraction</strong></td>
<td>43</td>
</tr>
<tr>
<td>4.1 Discarded feature-extractors</td>
<td>45</td>
</tr>
<tr>
<td>4.2 Partition functions and relational features</td>
<td>47</td>
</tr>
<tr>
<td>4.3 QRS-morphology</td>
<td>52</td>
</tr>
<tr>
<td>4.4 Wavelet transform</td>
<td>56</td>
</tr>
<tr>
<td>4.5 Higher order spectra</td>
<td>69</td>
</tr>
<tr>
<td>4.6 Auto regressive models</td>
<td>77</td>
</tr>
<tr>
<td>4.7 RR-interval features</td>
<td>77</td>
</tr>
<tr>
<td>4.8 The feature database</td>
<td>79</td>
</tr>
<tr>
<td><strong>5 Feature Selection</strong></td>
<td>83</td>
</tr>
<tr>
<td>5.1 The taxonomy of feature selection</td>
<td>84</td>
</tr>
<tr>
<td>5.2 Feature ranking by inference testing</td>
<td>87</td>
</tr>
<tr>
<td>5.3 Feature subset search in the multi-class setup</td>
<td>94</td>
</tr>
<tr>
<td>5.4 Novel methods on multi-class separation</td>
<td>98</td>
</tr>
<tr>
<td><strong>6 Core Classifier</strong></td>
<td>107</td>
</tr>
<tr>
<td>6.1 Choosing a classifier</td>
<td>108</td>
</tr>
<tr>
<td>6.2 The support vector machine</td>
<td>111</td>
</tr>
<tr>
<td>6.3 Hyper-parameter-search via optimal histograms</td>
<td>123</td>
</tr>
<tr>
<td>6.4 Multi-class support vector machines</td>
<td>131</td>
</tr>
<tr>
<td>6.5 The final setup</td>
<td>136</td>
</tr>
<tr>
<td><strong>7 Results</strong></td>
<td>139</td>
</tr>
<tr>
<td>7.1 SVM setup determination</td>
<td>141</td>
</tr>
<tr>
<td>7.2 Massive feature selection works</td>
<td>150</td>
</tr>
</tbody>
</table>
7.3 Performance of Feature Extractors and normalizations 161
7.4 The final feature set 168
7.5 Current ECG-classifiers lack from reliability and performance 170
7.6 The myth of the perfect wavelet 173

8 Conclusion 179
8.1 The current frontier of research 180
8.2 Modern feature extraction 180
8.3 Fast and high performing multi-class feature selection 182
8.4 New insights in the application of support vector machines 183

Appendices iii

A The final feature sets iii
A.1 The TopPCCorr(2, 50) dataset (only SWT features) iii
A.2 The TopPCCorr(4, 0.5) feature-set (only SWT features) iv

B The confusion matrices vii

C The scoring of this thesis’ feature sets xi

D Summary of reviewed publications xiii

E Detailed scoring of reviewed publications xv

Bibliography xxvii

Acknowledgement xlvii
Abstract

The electrocardiogram (ECG) is the most central information provider for the diagnosis of cardiac diseases. Today, the computer-aided analysis of short-term ECG-recordings in supine rest, the OR or the ICU is a well-established procedure. One major problem, though, is to track the presence and the amount of arrhythmias over days, weeks and month. As cardiologists can not spend their time on analyzing millions of heart beats for a single patient, an automated computer algorithm is necessary. Those ECG-classifiers assign heart beats to their respective type of arrhythmia or heart rhythm type.

Analyzing the current state-of-the-art is obligatory, but quite cumbersome. Performance measures can be misleading. Data is sometimes collected to the advantage of certain classifiers. Therefore, this thesis calculates the same performance criterion for all publications and assesses the likelihood for the reported performance to be achieved in real-life situations. For the purpose of the latter, this thesis introduces the \( \rho \)-Score which rates each classifier towards eight different aspects by qualitative and quantitative criteria. The final review of 72 ECG-classifiers presents new insights towards the current frontier of research in the field.

This thesis transcends this border by exploiting over 1.4 million features incorporating features derived from common methods like wavelet transformation, time-domain analysis, autoregressive models and from novel ones like the 2D-segmentation of the bi-spectrum, static delineation and heavy use of partition functions and relational features. In contrast to current practice, this thesis extracts its features from three different segments using several normalization strategies instead of a single one, fixed with respect to size and position.

However, those 1.4 million features have to be reduced to a set of 20 to 60 highly relevant ones. The most potent strategies for this amount of data reside in the text-categorization domain. Unluckily, those techniques are neither geared to process continuous real valued data nor to account for multi-class classification tasks. That is why this thesis extends those strategies to cope with cardinal data by creating new measures of effect size and novel strategies of combining them with regard to a multi-class scenario.

The final classifier employs support vector machines. During the course of the thesis, over 200,000 well-configured SVMs had to be calculated for certain experiments, necessitating 400 configurations per SVM to be tested. This number could be reduced to 7 by revealing a yet undiscovered connection between the optimal kernel-band-width for time-histograms and the optimal configuration of non-linear SVMs, saving much computation time.
Kurzfassung

Seit Jahrzehnten dient das Elektrokardiogramm, kurz EKG, der Diagnose von Herzkrankheiten. Sowohl im Operationssaal als auch auf der Intensivstation helfen schon heute computergestützte Analysen des EKGs die Therapie der Patienten zu verbessern. Bisher werden jedoch vor allem die Herzrate und nur auf kurzen Zeitskalen die Morphologie der einzelnen Herzschläge untersucht. Um die Morphologie der Herzschläge auch über Tage, Wochen oder gar Monate zuverlässig bewerten zu können, sind hochleistungsfähige EKG-Klassifikatoren notwendig, welche jedem Herzschlag automatisiert einen bestimmten Typ zuordnen.


Die geeignetsten Verfahren zur Merkmalsselektion für diese Menge an Daten stammen aus dem Bereich der Textkategorisierung (z.B. Spam-Filter). Unglücklicherweise sind diese Techniken nicht für kardinale Daten ausgelegt und funktionieren vor allem bei binären Problemen (Spam bzw. kein-Spam). Daher erweitert diese Arbeit die Textkategorisierungstrategien auf reelle Daten und schlägt Methoden vor, um diese in meinem multiklassen Szenario (normaler Herzschlag, Arrhythmie 1-12) einsetzen zu können.

Die endgültige Klassifikation geschieht mittels Support Vector Machines. Im Verlauf der Arbeit mussten für die verschiedensten Experimente insgesamt über 200.000 optimal konfigurierte SVMs erstellt werden. Jede einzelne SVM müsste normalerweise mit ca. 400 Konfigurationen getestet werden. Diese Zahl konnte auf 7 und die Gesamtrechenzeit deutlich reduziert werden. Dies war möglich durch dass Ausnutzen eine bislang unbekannten Zusammenhangs zwischen der optimalen Kernel-Bandbreite bei Zeithistogrammen und der optimalen Konfiguration der nicht-linearen SVM.
Chapter 1

Introduction

The electrocardiogram (ECG) is the most central information provider for the diagnosis of cardiac diseases and is widely used for performance assessment in sports and health care. In the last three decades great efforts have been made to assist cardiologists and nurses with pieces of computer technology. The major concern was to ease the measurement of the ECG during surgery or other procedures and to analyze the morphology of the QRS-complex in short ECG recordings in supine rest.

In the early days of scientific computing the first attempts were made to translate decision-making processes of a clinician into hard or fuzzy rules applicable for a computer [1]. Two decades later, computing performance was high enough to attempt to emulate cardiologists’ reasoning by decision trees [2]. In the nineties, ECG-classifiers flourished and big improvements to the quality and reliability of those were accomplished. As McLaughlin et al. [3] concluded in 1996: "Automated analysis of the electrocardiogram is a well-established procedure, with the initial impetus coming from active research groups in the early 1960s (1, 2).” Therefore short-term ECG-analysis is, in part, a closed case.

In the late nineties, Silipo and Marchesi [4] formulated a new goal for the subsequent decade:

"Thus, designing low-cost, high-performance, simple to use, and portable equipment for electrocardiography, offering a combination of diagnostic features, seems to be a goal that is highly worthwhile. Such equipment should embed and integrate several techniques of data analysis, such as signal processing, pattern detection and recognition, decision support, and computer interaction."


But the advances in remote monitoring and capabilities to store large amounts of data in small devices allow the investigation of even more interesting, yet unanswered questions: How does the amount of different arrhythmias (nodal, junctional or ventricular premature/escape beats) change over time in certain patients? What is the distribution of rare arrhythmias in the normal population? What is the weekly course of those arrhythmias? And furthermore, is heart variability as a measure of autonomous nervous balance capable of having a real impact in daily practice.

The long-term recordings necessary to answer those questions have a vast amount of data which can not be processed by a physician. Unfortunately, classifiers developed for short-term analysis in constraint environments won’t perform well in this long-term scenario. Noise due to baseline-wander, muscle noise, movement artifacts, signal disconnects and so forth pose a severe threat to those classifiers. Moreover, automated long-term recording diagnosis is heavily affected by misclassifications. In short recordings
of around 100 beats, 95% accuracy yields 5 misclassifications, which are potentially detected by the physician. In long-term recordings one forfeits the physician as a safety net. The heartbeats 100,000 times on a typical day. Therefore, such a classifier would yield 5000 misclassifications. This would render the aforementioned investigations on arrhythmia distribution infeasible.

That is why the first task in this thesis was to review the current state of the art of ECG-diagnosis. It defines a new Reliability Score. This $\rho$-Score assesses the likelihood of good real-life performance of the classifier by quantitative and qualitative criteria. This thesis shows that the most accurate performance description, the $j\kappa$-index (proposed by [7]) is anti-correlated with the $\rho$-Score. In essence, more reliable classifiers (high $\rho$-Score) give a more realistic estimate of their real-life performance, which is therefore worse (low $j\kappa$-index) compared to less reliable ones. In reverse, high-performing classifiers have a low reliability. The $\rho$-Score assessment allows the quantification of this relation and reveals the current frontier of reliability and performance each new classifier has to transcend.

The second task of this thesis was to exceed this boundary. To accomplish this task, one seeks to find the most accurate and concise description of the ECG and its idiosyncrasies. In the literature dozens of feature extraction and transformation techniques have been tested and evaluated for the purpose of ECG-classification. Nonetheless feature extraction methods are incomparable by means of classifier performance. This is due to preprocessing and classification strategies being so vast that one can’t argue for any feature-extractor being a priori superior, judging only by classifier accuracy. Good features are potentially obscured by improper preprocessing or low performing core classifiers. On the other hand, bad features can shine in the light of elaborate preprocessing or novel high-performing classifiers.

This thesis will show that it is possible to generate over 1,400,000 features from different extractors and find a proper multi-class feature selection strategy which yields stable, non-random and high-performing feature sets. The advances in multi-class feature selection shown in this thesis allow the comparison of different feature-extractors and to assess the impact of different normalization strategies. Give the current state of research, this thesis provides the most extensive analysis of feature-extractors of the ECG.

Moreover, this thesis will refute the myth of the perfect wavelet. Many researchers have strived for finding a wavelet that perfectly describes the ECG within the framework of orthogonal discrete wavelet transforms. The major problem with this approach is that different types of arrhythmias have different time-frequency characteristics. Therefore, a single orthogonal wavelet is not likely to describe them. The thesis shows that either one has to relax the orthogonality constraint on the wavelet or use many orthogonal wavelet transforms and derived parameters.

Furthermore, this thesis will argue why the developed multi-class feature selection technique is superior to state-of-the-art multi-class selections strategies.

In the end, a good feature set is useless without a proper core-classifier, which I choose to be support vector machines. The development of the final classifier took thousands of pre-evaluations, testing and feature set assessments. Therefore, computation time was a limiting resource. This necessitated a speed up of the whole training process. In this thesis, a novel method was developed that increases the speed of multiclass-training four-fold. Moreover, this technique gives direction to a solution for the best-suiting-kernel task. A major question when employing support vector machines is which kernel to use. The answer this thesis suggests is: The optimal kernel optimizes the sampling of the histogram of the distance of the high dimensional feature vectors.

After these motivational remarks, this introductory chapter will be outlined as follows: First, I give a short outline of the thesis at hand which is followed by the definition of the mathematical framework used herein. Afterwards I will explain the framework for judging performance and reliability of the proposed
classifier: the Reliability Score (\( \rho \)-Score). The \( \rho \)-Score developed in this thesis will later on be used in the review of current state-of-the-art ECG-classifiers. In the fourth section, I will explain basic properties of the data used within the thesis. The concluding section will show the physiologic and electric principles that enable the measurement of the ECG. Furthermore, the section will demonstrate the high noise and interpatient-variability of heartbeats within the data, which is addressed in the subsequent chapter: Preprocessing.

**Outline**

The outline of this thesis will be built up like comparable work [8], consisting of five major chapters: preprocessing, feature extraction, feature selection, classification and results. After this introductory chapter, I will explain the preprocessing of the ECG-signal in detail, i.e. filtering noise and artifacts, detection of the QRS-complexes and segmentation of the ECG. Subsequently the feature extraction chapter will describe the different features used for the characterization of the arrhythmias the ECG-signal contains. It shows in detail the "static delineation" approach, the asymmetric QRS-width measures, the higher-order statistics measures and the asymmetric principal component analysis developed in this thesis. Moreover, it will give an insight in modern wavelet transformations and will argue in favor of using features from different wavelet-families, rather than engaging into the search for the "perfect" wavelet.

After the features are defined, the feature selection chapter will demonstrate why feature selection in a setup of around 1,400,000 features will not work with standard wrapper methods and that a new feature selection method gives promising results using boot-strapping techniques to guard against implicit overfitting.

In the adjacent chapter concerned with the classification-process, the author will argue for the use of support vector machines (SVM) as the core classifier of the classification task at hand. Moreover, this chapter will introduce a novel optimization scheme for the determination of the kernel parameters of the SVM that can find the optimal parameters faster and with comparable accuracy. The approach presented in this thesis gives direction to finding the optimal kernel for SVM under the given dataset. Additionally, the chapter will present a novel method for semi-supervised training set selection, i.e. to select a training set that increases the reliability of the classifier at hand. The last chapter will show the results of the novel methods developed in this thesis.
Chapter 2

ECG-Basics

This chapter is intended to enable the reader to understand the environment in which ECG-classification takes place. Surely, everyone has seen an electrocardiogram whether it has been one of a friend, family member or of oneself. However, little is known about why it looks the way it is and what the physics are behind an ECG-measurement. In short, the ECG measures the integrated electric field of Hertz-dipoles evolving in the course of the contraction and relaxation of the heart muscle. The first section will give a more in-depth analysis of how the ECG is shaped by the in- and outflux of ions in tiny pacemaker and muscle cells. The second section will focus on real-life ECGs that are utilized in this thesis. It presents the data-material for the ECG experiments and the machine-learning databases employed for verifying and testing the configuration of algorithms explained later on. It discusses the limitations and constraints of the ECG data available and shows examples for the arrhythmias that are the subject of the ECG-classifier.

The third section will transfer the ECG into the mathematical domain. In the analog world, the ECG can be considered a continuous stream of information. In the digital world though, the ECG is sampled at discrete times yielding a simple time series with equidistant sampling. The third section presents the mathematical definitions used throughout this thesis and can be considered as a brief version of this thesis. It shows the way from the sampled time series to the final classification of the ECG via support vector machines.

The first three sections reflect on the physics, biology, and maths perspective of ECG-classification. The fourth section will conclude this chapter presenting the medical or practical point of view. For a physician, the major question is: Which ECG-classifier will turn out to be robust and high-performing in a real-life application? More precisely: How does the performance reported in a certain publication relate to its real-life performance? The $\rho$-Score assessment, developed in this thesis, addresses those questions. It rates several aspects of an ECG classifier like data usage, performance measures utilized, number of features and so on.

2.1 The origin of the ECG

This section is intended to enable the reader to have a basic understanding of the measurement of the activity of the heart muscle and how it is reflected within the ECG. First, I will present the time-course of a single contraction and relaxation cycle of the heart muscle. In this subsection, it is sufficient to comprehend that during one heartbeat the heart-muscle cells get “electrically activated”, i.e. their cell-membrane depolarizes which means that the membrane-potential turns from negative (ground state) to positive. This electrical excited state or action potential propagates from the atria to the ventricles and causes their contraction and relaxation (repolarization). Obviously, the electrodes of the ECG-device on
the skin can not measure those membrane-potentials. Therefore, the second subsection will demonstrate how action potentials cause a local-separation of anions and cations in the extracellular vicinity of each cell. This creates micro dipoles which trail along the action potentials. The integration of all those dipoles and their corresponding electric fields is what the ECG really measures. At the end of the section, ECG segments originating from three different heart beat morphologies are presented.

This section is based on two text books: [9] (physiology, anatomy and membrane potential) and [10] (dipole theory).

2.1.1 The course of the ECG

From a simplistic approach the ECG is generated as follows: The sinus node, the generator of the sinus-rhythm, which is located near the entrance of the superior caval vein starts the heartbeat cycle. The cells within the sinus node depolarize, i.e. the membrane potential of the cells becomes positive. As cells in the heart are connected via gap-junctions with low resistance, nearby cells depolarize, too. This atrial depolarization is reflected within the ECG as the P-wave, which is depicted in fig. 2.1. The atria and the ventricles are electrically isolated from each other. Therefore, the activation potential has to travel through the atrio ventricular node (AV-node), which acts like a bottleneck, decreasing propagation speed. The time gained is used for filling the ventricles. After passing the AV-node, the AP travels through the Purkinje fibers and depolarizes the whole ventricle. The contraction of the ventricles at the R-peak (see fig. 2.1) is followed by a slow repolarization of the ventricles, which is reflected by the T-wave.

![Diagram of the ECG](image)

This is the short version. One now may ask: How do small current changes in cells manifest in electrodes on the body surface? Why is the ECG dropping at the Q-point? Why is the T-wave negative, when there
2.1 The origin of the ECG

is a repolarization? The next paragraphs will answer those questions by introducing the two major concepts needed to understand the formation of the ECG: The action potential and the dipole theory. (see [10] for a complete derivation)

2.1.2 The action potential

The origin of the normal sinus-rhythm are the cells of the sinus node. The membrane of those cells, as all cells in the body, have a negative resting potential. It is the result of two major influences: charge and concentration gradients between the cells interior and the extracellular matrix. Whereas sodium-, chloride- and calcium-ion concentrations are low within the cell, their concentration is high in the extracellular space. The situation is reversed in case of hydrogen and potassium ions. In fact, depolarization of the cell, i.e. the increase of the membrane potential, is mainly governed by sodium resp. calcium ion-influx. The major driver of repolarization on the other hand is the potassium outflux. The formation of the action potential AP is as follows: First an influx of sodium-ions increases membrane potential. As a result, voltage-dependent channels open up and increase sodium-permeability of the membrane. The membrane potential rises until it overshoots and sodium-selective channels close. In the aftermath of the overshoot, potassium-selective channels open and the outflux of the potassium-ions repolarizes the membrane potential. The reestablishment of the resting potential is achieved via the sodium-potassium pump. It flushes three sodium ions out and suck two potassium into the cell. As this transport-process is performed against the concentration gradient, it necessitates energy, which is provided by the body's fuel: Adenosin-Tri-Phosphate (ATP).

The action potential (AP) of the sinus node is now easily created as the depolarization of one cell is mediated via the gap-junctions to other cells in the vicinity. The AP propagates like a wavefront: At the frontier, cells get depolarized and behind the wavefront cells are in rest, waiting for the next AP to depolarize them.

2.1.3 The dipole theory

Now let us take a closer look at the frontier of the AP's wavefront. With regard to the extracellular space, repolarization and depolarization are reversed during the AP. It is straight forward: while the cytoplasm depolarizes, the extracellular space repolarizes and vice versa for the cytoplasms repolarization. Let us say the wavefront travels from right to left. On the left side of the wavefront cell membranes are at resting potential, causing the extracellular space to be locally depolarized (positively charged). On the right side of the wavefront, on the other hand, it is locally repolarized as cell-membranes are still depolarized. The positively charged left side of the wavefront and the negatively charged right side of the wavefront form a dipole. The dipole propagates in the direction of the AP with its dipole moment being parallel to the propagation vector.

2.1.4 Measuring the dipole

Obviously, the ECG does not measure the local dipole moment. Instead, it measures the integrated electric far-field of all local dipoles. The magnitude of the field depends on the number of local dipoles and therefore on the number of depolarizing cells at the AP's wavefront. The direction of the electric field is the net vector of all local dipoles summed together. This electric field manifests itself in the different leads of the ECG. The orientation of the lead determines the magnitude of the voltage measured at the
respective lead. Let us think about a typical 3-lead system according to Einthoven (lead I, II and III). The limb lead II measures the potential difference between the foot and the right shoulder electrode. Let us imagine a straight copper wire connecting the two electrodes. The voltage measured between those electrodes (foot electrode as the positive one) is proportional two the map of the vector of the electric field onto the direction of the copper wire.

Providing the connection line of the two electrodes is parallel to the electric field, the voltage between those electrodes is maximal. If the connection line is perpendicular to the electric field, no voltage can be measured. This observation is consistent with the dipole theory. A dipole does not emit radiation perpendicular to its dipole moment. The origin of those dipoles can be easily deduced if endocardial and many body surface measurements are available. The major challenge is to determine the class and origin of an ECG-beat by only one or two channels.

### 2.1.5 Morphologies

Figure 2.2 show typical examples of the ECG-segments of normal heart beats (N-beat), ventricular premature contractions (VPCs) and left bundle branch blocks (LBBB). Each heart beat is presented as a 300 samples sized segment centered around the R-peak. Those beats originate from the modified lime lead II in the MIT-BIH Arrhythmia database.

The N-beats show the characteristic features explained in the standard scheme (see fig. 2.1): a pronounced QRS-complex which is preceded by the P-wave and the subsequent T-wave. As explained in previous subsections, the primary excitation propagates from the sinus node into the ventricles.

This process can be superseded by spontaneous foci of electrical activity in the ventricles itself. The action potential can spread from those foci over the whole ventricle causing premature contractions (VPCs) which precede the normal sinus rhythm. This is the major reason why ECG-segments in the middle panel of fig. 2.2 lack a prominent P-wave but contain smudged QRS-complexes. The VPCs are not as homogeneous as the N-beats and the LBBB, as their shape is governed by the location of the foci in the ventricle.

LBBBs occur in case the AP's propagation in the ventricles is asynchron with respect to the left and right ventricle. If the left bundle branch is blocked, the excitation of the left ventricle is delayed, which is reflected by the “high-shoulder” of the QS-segment.

### 2.2 Data material

This section describes the different databases that were used in this thesis. In brief: The MIT-BIH Arrhythmia database was used as the gold standard for the assessment of ECG-morphology classification. A second database, the Longterm ST-Database, was used to evaluate the effect of implicit over-fitting towards the selected ECG-features. The ECG material was provided by the Physionet-Project.

In this thesis, an ensemble of support vector machines (SVM) is used for the final classification. However, the configuration of the SVMs is crucial. There are different data normalization techniques, several ensemble combination schemes, training criteria and SVM hyper-parameters to be optimized. This thesis will use a recent database provided by Galar et al. to test several SVM configurations on non-ECG data. This database is also used, together with some datasets from the UCLA machine learning database,
to prove that the new SVM hyper-parameter search, developed in this thesis, yields high performance in less computation time.

2.2 Data material

2.2.1 The MIT-BIH Arrhythmia database: The gold-standard

The main target for classification is the MIT-BIH Arrhythmia database. It is widely used for assessment of ECG-classifiers as already described in section 2.4. Therefore, this database was used for all preliminary results, training and validation of ECG-feature selection, preprocessing and classification. The MIT-BIH Arrhythmia database consists of 48 datasets all annotated by ECG-experts. As this thesis focuses on single-channel ECG classification, two datasets (“102” and “104”) are discarded as they do not use the modified limb lead II (MLII). The amount of available annotated beats is shown in tab. 2.1. Moreover, it presents the number of beats available after preprocessing and QRS-detection.

<table>
<thead>
<tr>
<th>shortname</th>
<th>DB-Annotation</th>
<th># of beats(a)</th>
<th>QRS-morphology type</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>. or N</td>
<td>75260 (74547)</td>
<td>normal beat</td>
</tr>
<tr>
<td>LBBB</td>
<td>L</td>
<td>8075 (8057)</td>
<td>left bundle branch block beat</td>
</tr>
<tr>
<td>RBBB</td>
<td>R</td>
<td>7259 (7245)</td>
<td>right bundle branch block beat</td>
</tr>
<tr>
<td>VPC</td>
<td>V</td>
<td>7124 (7091)</td>
<td>premature ventricular contraction</td>
</tr>
<tr>
<td>vP</td>
<td>/ or (P)</td>
<td>3652 (3609)</td>
<td>paced beat</td>
</tr>
<tr>
<td>APC</td>
<td>A</td>
<td>2546 (2532)</td>
<td>atrial premature beat</td>
</tr>
<tr>
<td>FVN</td>
<td>F</td>
<td>803 (801)</td>
<td>fusion of ventricular and normal beat</td>
</tr>
<tr>
<td>VF</td>
<td>'[]'</td>
<td>490 (—)</td>
<td>beats with ventricular flutter</td>
</tr>
<tr>
<td>MISS</td>
<td>M or MISSB</td>
<td>428 (—)</td>
<td>missed beat</td>
</tr>
<tr>
<td>FPN</td>
<td>f</td>
<td>260 (260)</td>
<td>fusion of paced and normal beat</td>
</tr>
<tr>
<td>NESC</td>
<td>j</td>
<td>229 (229)</td>
<td>nodal junctional escape beat</td>
</tr>
<tr>
<td>AAP</td>
<td>a</td>
<td>150 (149)</td>
<td>aberrated atrial premature beat</td>
</tr>
<tr>
<td>VESC</td>
<td>E</td>
<td>106 (106)</td>
<td>ventricular escape beat</td>
</tr>
<tr>
<td>NPREM</td>
<td>J</td>
<td>83 (83)</td>
<td>nodal junctional premature beat</td>
</tr>
<tr>
<td>OTHER</td>
<td>not</td>
<td>806 (84)</td>
<td>all other beats in the database</td>
</tr>
</tbody>
</table>

Table 2.1: Statistics of the different QRS-morphologies provided by the MIT-BIH Arrhythmia database with respect to the MLII lead.

(a) after QRS-detection and Noise-reduction

The amount of beats changes due to imperfect QRS-detection. The QRS-detection presented in chapter three is not able to detect ventricular flutter (VF) and missed beats (MB). Moreover, only a small portion of OTHER-beats is detected. Further reduction of beats is caused by the great length of the FIR-bandpass-filter used for noise reduction. Three to five beats at the end of each record are discarded, due to the group delay of the filter. All records in the database have been sampled at 360Hz. The ECG-amplitudes cover a range of 10mV at a resolution of approx. 0.005mV.

**Prasad and Sahambi [17] training and test split** The amount of data used for training and testing is chosen according to Prasad and Sahambi [17]. The number of samples available is highly skewed with respect to the 15 arrhythmias. Hence, only 25% of the normal beats and one third of the VPCs, the left and right bundle branch blocks are used for training. All other morphologies contribute 50% of their samples to the training. Overall, 28.8% of the data is used for training.

1 Christov and Bortolan [16] try to separate VPCs from non-VPCs. They could not demonstrate, that using two leads is superior to one lead.
2.2.2 The Long-Term ST Database: testing implicit over-fitting

As already said, the whole preprocessing, feature selection and feature extraction is guided by results gained in classification of the MIT-BIH Arrhythmia datasets. This procedure somehow leads to an implicit over-fitting, i.e. features and preprocessing techniques are adjusted to the noise, artifact and movement level of the ECG-records in the database.

Implicit over-fitting can be understood in the following way: Let us think of fitting experimental data of 100 sample points by a function yet to be determined. One can test dozens of functions with less than 4 parameters which will not cause the same explicit over-fitting one would get, if one, for example, fits those 100 sample points with a polynomial of degree 100. The latter approach will yield a perfect match to the data all the time. So, restricting the fitting-parameters below 4 will guard against parameters that are too focused on the experimental data sampled. When testing many functions for fitting, one implicitly introduces a new category of fitting parameter: the fitting function itself.

Multi-hypothesis testing This implicit over-fitting is directly related to the notion of multi-hypothesis testing (MHT). Assume a statistical hypothesis test yielding a p-value, i.e. measuring the probability that the null-hypothesis (data can not be fitted) is true, but one refuses the null hypothesis, favoring the alternative hypothesis (data can be fitted). p-values below 0.05 indicate that it is most likely that the fit did converge and the data can be fitted well using the fitting function at hand. The p-value can be regarded as an upper bound of the false positive rate which assesses the likelihood that a positive test-result in the sampled data will hold for the parent population.

MHT goes one step further and asks: If one tests many hypothesis (fitting functions) on that certain data material, what is the probability that one gets a significant p-value by chance? Whereas the p-value measures the false positive rate, MHT asks for the false discovery rate. A new methodology introduced by Storey [18][19] converts the traditional p-value to the q-value, which can be understood in the following way:

Figure 2.2: ECG segments of 300 samples length originating from the MIT-BIH Arrhythmia beats for three heart beat types: N-beat (top), VPC (middle), LBBB (bottom).
Let us suppose from 1000 fitting functions, 100 will yield \( p < 0.01 \). After calculating the q-value the highest of the 100 p-values converts from say 0.01 to 0.1. This tells us that even though the false positive rate is around 0.01, ten out of those 100 p-values will be false discoveries. Nothing can be told if a certain fitting function is in fact a false discovery, but choosing low q-values guards for high amounts of false discoveries.

Going back to the topic of classification one will find that testing the performance with dozens of feature sets, preprocessing parameters etc. leads to an infinite number of statistical hypothesis tests that no one can carry out, not to mention calculating p or q-values. Because typical MHT can’t be performed, one has to find another verification set that will be tested only at the end of this study.

Database characteristics A major obstacle in finding a suitable database is that the MIT-BIH Arrhythmia provides a great variety of different morphologies which are not present in other databases. Moreover, the MLII lead is not widely used by other databases and the sampling frequency of 360Hz is not standard. From the author’s point of view, there is no second database with a comparable distribution of ECG morphologies based on the MLII lead.

That is why one has to seek for a compromise. The Long-Term ST database (LTST) provides much data and beats measured employing the “ECG”-lead are comparable to those measured using the MLII lead. Thus, it should be possible to use the feature set calculated on the MIT-BIH Arrhythmia database for the assessment of the LTST-database. The LTST-database is used to perform a three-class classification: Normal vs. VPC vs. OTHER heart beats. A more detailed analysis is not possible, because LTST does not provide more morphologies with at enough heart beats.

In section 2.4 I will stress that with current databases it is only possible to either do record-based classification or beat-based classification. The latter uses all records and randomly assigns beats to either a training or a test set. This is necessary to generate a bootstrap estimate for the accuracy of the results. On the other hand all records are partly used during training. The record based approach does lack this problem. Instead, it lacks the possibility to compute a bootstrap estimate, as records can not easily rearranged in either the training or test-set. Doing so would change the distribution of the number of heart beats in the training and test-set with respect to different morphologies.

Hence, this thesis is going to use beat-based classification for the MIT-BIH Arrhythmia database and employs record-based classification for the LTST database. The statistics of morphologies in either the training or test-set for the evaluation of the feature-sets with the LTST database are given in tab. 2.2.

The sampling frequency of the database is only 250Hz. Consequently, the ECGs are resampled to 360Hz in order to match with the MIT-BIH Arrhythmia database. The voltage-resolution is also 0.005mV but at a range of ±10mV.

<table>
<thead>
<tr>
<th>shortname</th>
<th># of beats in training-set</th>
<th># of beats in test-set</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1089960</td>
<td>1104455</td>
</tr>
<tr>
<td>VPC</td>
<td>9565</td>
<td>11979</td>
</tr>
<tr>
<td>OTHER</td>
<td>3000</td>
<td>5497</td>
</tr>
</tbody>
</table>

Table 2.2: Statistics of the different QRS-morphologies provided by the Long-Term ST database with respect to the “ECG” lead.

\(^{a}\) records: 20031, 20251, 20271, 20273, 20281, 20351, 20371, 20391, 20411, 20431, 20451 \(^{b}\) records: 20041, 20061, 20272, 20274, 20341, 20361, 20381, 20401, 20421, 20441, 20461
2.2.3 ECG-lead selection

As we will see later on in the section about the morphology of the ECG, the selection of the lead measuring the ECG is crucial. Even though the number of publications employing multi-lead approaches increases LLamendo and Martinèz [20, 21], Barro et al. [22] the vast majority of detectors and classifiers focus on a single lead. In fact, most of the algorithms are tested on the MIT-BIH Arrhythmia [12] database. That is why most of the publications [23, 24] concentrate on the modified limb lead MLII, which is the predominant lead in the database\(^2\). The latter arguments are solely pragmatic in nature. Actually, free access to annotated multi-lead ECGs of longer duration and of quality comparable to the MIT-BIH Arrhythmia database [12] would be of great benefit to the scientific community. The "simultaneous processing of various signal channels"[22] has several advantages: "A ventricular beat, for example, may present a practically normal morphology in some channels, while being clearly aberrant in others."[22] Moreover, with more channels the immunity to noise can be improved[22]. As aforementioned, the MIT-BIH Arrhythmia is not well suited for this analysis, because there are not enough recordings that include the same two leads. So one has to focus on the MLII lead only.

This is quite problematic. The MLII lead is not the prevailing lead in other databases. Therefore, one can not train the classifier using the MIT-BIH Arrhythmia database and cross-validate it, in case one wants to assess all 13 arrhythmias. In conclusion, the power of the feature sets that will be selected during the course of this thesis, will be valid only on the MLII lead. The final test will be performed on the "ECG" lead of the Long Term ST database. It shows morphologies comparable to those in the MLII lead.

Cross validation is also aggravated due to the gain resolution and sampling frequency of the different ECG-Holter systems. Whereas the MIT-BIH Arrhythmia uses data sampled at 360Hz \(^3\) with a 11-bit resolution (±5 mV), the Long Term ST-database uses voltages sampled at 250Hz with a 12-bit resolution (±10 mV).

2.2.4 Machine learning databases

One major aspect of this thesis is to improve the hyper-parameter search in support-vector machines. Clearly, superiority of the proposed strategy should be assessed on comparable datasets. This thesis will, therefore, utilize the datasets employed by [26] and in part by [27], who also strived for a faster hyper-parameter search scheme. The table below gives the basic properties of those datasets.

<table>
<thead>
<tr>
<th>dataset</th>
<th>#features</th>
<th>(N_{\text{train}})</th>
<th>(N_{\text{test}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>banana</td>
<td>2</td>
<td>400</td>
<td>4900</td>
</tr>
<tr>
<td>diabetes</td>
<td>8</td>
<td>468</td>
<td>300</td>
</tr>
<tr>
<td>image</td>
<td>18</td>
<td>1300</td>
<td>1010</td>
</tr>
<tr>
<td>splice</td>
<td>60</td>
<td>1000</td>
<td>2175</td>
</tr>
<tr>
<td>ringnorm</td>
<td>20</td>
<td>400</td>
<td>7000</td>
</tr>
<tr>
<td>twonorm</td>
<td>20</td>
<td>400</td>
<td>7000</td>
</tr>
<tr>
<td>waveform</td>
<td>21</td>
<td>400</td>
<td>4600</td>
</tr>
</tbody>
</table>

Table 2.3: Properties of seven datasets from the UCI Machine Learning Repository [28] used for binary classification tests: dataset name, number of features, number of training samples \(N_{\text{train}}\) and number of test samples \(N_{\text{test}}\)

\(^2\) All records but record "102" and "104" were measured using the MLII-lead. \(^3\) In the early eighties and seventies such high sampling rates, caused severe problems to the computer architecture, because one 24h-ECG would have a raw size of 100 MB with single precision and 200MB with double precision. These difficulties urged for data compression techniques like [25].
2.3 Mathematical framework

The major reason for employing those sets is that [29] provided 100 randomized training and test splits and put them on the Internet. Hence, one can compare methods on exactly the same data. As comparability is a key issue, the other three datasets in [26, 27] (adult, web, tree) are discarded from the investigation. [26, 27] do not provide any possibility to generate the original data of their experiments. Clearly, seven datasets only give a first impression of the performance of hyper-parameter selection schemes.

A more detailed analysis is carried out using a multi-class benchmark developed by [15]. They provide 18 multi-class datasets, which are split into five training and test splits, allowing to perform a five fold cross-validation. Even though Galar et al. [15] coin their method "cross-validation" they perform a boot-strap estimate of the data. The data-folds are stratified and of equal data amount, i.e. in each fold the number of samples corresponding to a certain class is constant. In this scenario, the boot-strap and cross-validation estimates of accuracy and other performance measures are the same. The basic properties of the datasets are given in tab. 2.4. The datasets also originate from the UCI Machine Learning Repository [28], but [15] generated five splits that can be retrieved from the Internet, which increases comparability.

<table>
<thead>
<tr>
<th>dataset</th>
<th>#features</th>
<th>nominal ones</th>
<th>N_data</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>6</td>
<td>0</td>
<td>1728</td>
<td>4</td>
</tr>
<tr>
<td>Lymphography</td>
<td>18</td>
<td>15</td>
<td>148</td>
<td>4</td>
</tr>
<tr>
<td>Vehicle</td>
<td>18</td>
<td>0</td>
<td>846</td>
<td>4</td>
</tr>
<tr>
<td>Cleveland</td>
<td>13</td>
<td>8</td>
<td>297</td>
<td>5</td>
</tr>
<tr>
<td>Nursery</td>
<td>8</td>
<td>8</td>
<td>1296</td>
<td>5</td>
</tr>
<tr>
<td>Page-blocks</td>
<td>10</td>
<td>0</td>
<td>548</td>
<td>5</td>
</tr>
<tr>
<td>Autos</td>
<td>25</td>
<td>10</td>
<td>159</td>
<td>6</td>
</tr>
<tr>
<td>Dermatolgy</td>
<td>33</td>
<td>32</td>
<td>366</td>
<td>6</td>
</tr>
<tr>
<td>Flare</td>
<td>10</td>
<td>10</td>
<td>1389</td>
<td>6</td>
</tr>
<tr>
<td>Glass</td>
<td>9</td>
<td>0</td>
<td>214</td>
<td>6</td>
</tr>
<tr>
<td>Satimage</td>
<td>36</td>
<td>0</td>
<td>643</td>
<td>7</td>
</tr>
<tr>
<td>Segment</td>
<td>19</td>
<td>0</td>
<td>2310</td>
<td>7</td>
</tr>
<tr>
<td>Shuttle</td>
<td>9</td>
<td>0</td>
<td>2175</td>
<td>7</td>
</tr>
<tr>
<td>Zoo</td>
<td>16</td>
<td>16</td>
<td>101</td>
<td>7</td>
</tr>
<tr>
<td>Ecoli</td>
<td>7</td>
<td>0</td>
<td>336</td>
<td>8</td>
</tr>
<tr>
<td>Led7digit</td>
<td>7</td>
<td>7</td>
<td>500</td>
<td>10</td>
</tr>
<tr>
<td>Penbased</td>
<td>16</td>
<td>0</td>
<td>1099</td>
<td>10</td>
</tr>
<tr>
<td>Yeast</td>
<td>8</td>
<td>0</td>
<td>1484</td>
<td>10</td>
</tr>
<tr>
<td>Vowel</td>
<td>13</td>
<td>0</td>
<td>990</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 2.4: Properties of 18 machine learning datasets provided by Galar et al. [15] for multi-class classification tests: dataset name, number of features, number of nominal features, number of overall data samples and number of classes to be distinguished

2.3 Mathematical framework

Many of the mathematical tools used in this thesis are not defined coherently in the literature. In this section, I will lay out the mathematical framework which is the foundation of definitions made later on. This should ease understanding and orientation and reduce confusion. Mathematical objects that have a dominant symbolic definition in the literature will be defined accordingly. If any definition coincides with the framework below, the exception from the framework will be marked promptly. The definitions carried out in the next lines follow the outline of the thesis from the raw-signal over preprocessing, to the feature extraction and selection stage, to the classifier and final result section.
**Raw signal** Let us first define the raw ECG-signal $y$ measured at sampling frequency $f_s$:

$$y_i = y(t_i) \quad i \in \{1, \ldots, N_{\text{raw}}\} \quad (2.1)$$

$$t_i = \tau_s \cdot i \quad (2.2)$$

$$\tau_s = 1/f_s \quad (2.3)$$

Given an ECG-record containing $N_{\text{raw}}$ sampled voltage amplitudes, the $i^{\text{th}}$ sample measured at time $t_i$ will be denoted as $y_i$. The distance between those measurements, denoted as $\tau_s$, is the inverse of the sampling frequency $f_s$ and is fixed for all records in the database at 360 Hz.

**Preprocessing** The untreated series of amplitudes is of minor benefit for ECG-classification because it contains high-frequency noise, artifacts of baseline-wander, interference with power line and electromyographic noise. Therefore, the signal $y$ is object to FIR filtering as described later on, yielding the filtered signal $y'$. But still, the processed signal $y'$ lacks information about QRS position and class membership. This issue is addressed by the subsequent segmentation. Its aim is to align all QRS-complexes onto their respective R-peak.\(^4\) Given the sample number of the R-peak of the $i^{\text{th}}$ beat $R_i$, the segmentation extracts $(\Delta_{\text{seg}} - 1)/2$ samples on the right and left side of the heartbeats R-peak.\(^5\) Those $\Delta_{\text{seg}}$ long segments form the raw beat matrix $Y(\Delta_{\text{seg}}) \in \mathbb{R}^{N_b \times \Delta_{\text{seg}}}$ according to eq. 2.4. The number of beats in the record is denoted as $N_b$.

$$Y(\Delta_{\text{seg}})_{ij} = y' \left( \left( R_i - \frac{\Delta_{\text{seg}} - 1}{2} + j - 1 \right) \cdot \tau_{\text{sampling}} \right) \quad j \in \{1, \ldots, \Delta_{\text{seg}}\} \quad (2.4)$$

A major problem within this matrix is that it incorporates intra and inter-patient variability. Both will be reduced by a normalization, described later on, which yields the final preprocessed beat matrix $Y'(\Delta_{\text{seg}})$.

**Class indexing** So far no information is provided about the class or label each beat is an instance of. The $i^{\text{th}}$ heartbeat $y_i$ is an instance of the $j^{\text{th}}$ class if eq. 2.5 holds.\(^\dagger\) $\iota_j \in \mathbb{R}^{N_j}$ is a vector containing the $N_j$ indices of the patterns representing the $j^{\text{th}}$ class. $N_j$ is the number of members in class $j$. The indices $\iota_j$ are retrieved from the annotation file of the corresponding ECG-database.

$$y_i \in \text{class } j \iff \exists k \in \mathbb{N} : i = \iota_{jk} \quad (2.5)$$

**Feature extraction** Since the preprocessing ensures a fair amount of intra-class variability, i.e. QRS complexes of a certain class are fairly comparable, the subsequent feature extraction can rely on benign QRS-complexes. Each feature-extractor can be understood as a map between the beats’ samples to a scalar according to eq. 2.6.

$$\phi : \mathbb{R}^{\Delta_{\text{seg}}} \rightarrow \mathbb{R} \quad (2.6)$$

The main feature set $\tilde{F}$ is formed by the set of all $F = |\tilde{F}|$ feature-extractors.

\(^4\) The position of the R-peak is defined according to the annotation in the raw ECG annotation file or by employing a QRS-detector (see section ...)

\(^5\) $\Delta_{\text{seg}}$ has to be odd to guarantee that the segment is symmetric around the fiducial point or R-peak
\[ F = \{ \phi_1, \ldots, \phi_k \} | k_i \in \{1, \ldots, F\} \]  

(2.7)

The features extracted from the preprocessed beats in \( Y'(\Delta_{seg}) \) will be stored in the feature matrix \( \Phi \in \mathbb{R}^{N_b \times F} \), generated using the feature set \( \tilde{F} \) according to eq. 2.8:

\[ \Phi = Y'(\Delta_{seg}) \odot \tilde{F} \equiv \begin{bmatrix} \phi_1(y'_1) & \phi_2(y'_1) & \cdots & \phi_F(y'_1) \\ \phi_1(y'_2) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \phi_1(y'_N) & \phi_2(y'_N) & \cdots & \phi_F(y'_N) \end{bmatrix} = [f_1, \ldots, f_F] \]  

(2.8)

Using the operator \( \odot \), each vector \( y'_i \) containing the preprocessed samples of the \( i \)-th beat will be transformed using the feature-extractors in \( \tilde{F} \). The raw feature matrix \( \Phi \) is comprised by \( N_b \) row vectors \( x_i \), identifying the \( i \)-th ECG-beat\(^6\). The feature vector of the \( i \)-th feature-extractor, denoted as \( f_i \), is the \( i \)-th column of the raw feature matrix \( \Phi \).

**Feature selection**  
In this thesis the final \( N_b \) equals 100,000 beats and the overall features \( F \) extracted exceed one million features. Clearly this amount of features is intractable even with modern computer technology. Therefore, feature selection is carried out, searching for \( \tilde{F}_{sel} \subset \tilde{F} \) with equal or better discrimination power and minimal redundancy of features. The thesis will follow a filtering strategy. Hence, each feature is rated according to a measure of discriminancy. This thesis will focus on effect-size indices as an estimate of discriminancy. They map two vectors \( v_1 \) and \( v_2 \) of potentially different size to a scalar value:

\[ \text{EFF}(v_1, v_2) : \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \rightarrow \mathbb{R} \]  

(2.9)

As this thesis investigates \( C = 13 \)-class feature selection methods, the filtering techniques are based on the interclass scoring table \( \Xi(\text{EFF}) \in \mathbb{R}^{F \times C \times C} \). It measures the power of a certain feature to separate patterns in class \( i \) from those in class \( j \) according to eq. 2.10.

\[ \Xi(\text{EFF}) : \Xi_{kij} = \text{EFF}(f_j^i, f_j^i) \]  

(2.10)

\( f_j^i \) denotes the class dependent feature vector of dimension \( N_j^i \):

\[ f_j^i : (f_j^i)_k = (y_{(i)_j})_k \]  

(2.11)

which stores the values of the \( i \)-th feature for all heartbeats in the \( j \)-th class.

The final challenge is to evaluate \( \Xi \) in order to extract \( F_{sel} \) features which are potent candidates for the classification stage. The final feature set generated by the feature selection is denoted as \( \tilde{F}_{sel} \).

---

\(^6\) in case of the machine learning databases \( x_i \) identifies the \( p \)-th pattern
**Classification** The feature selection reduces the dimensionality of the raw feature matrix $\Phi$ by removing the discarded columns (not selected features). This removal of columns yields $\Phi_S$ which is now object to classification.

Before the patterns of the $C$ different morphologies are assigned to their respective class, a proper normalization-transform has to be applied to $\Phi_S$. This ensures numerical comparability of the feature vectors. More precisely, each feature vector is of the same importance by means of magnitude. Henceforth, the processed feature vectors will be denoted as $f'_i$. They form the columns of the normalized feature matrix $\Phi'$. Its row vectors $x'_i$ still represent the features of the $i^{th}$ beat, but now with features that are normalized with respect to the features' statistical distribution.\(^7\)

The processed feature matrix $\Phi'$, especially its pattern vectors $x'_i$ will be the input of a general classifier $\Upsilon$, e.g. a support-vector-machine. The classifier is defined as a surjection between the feature vector space and the classification set according to eq. 2.12.

$$\Upsilon : \mathbb{R}^F \to \{1, \ldots , C\} \subset \mathbb{N}^+ \quad (2.12)$$

$\Upsilon$ yields the class-number of the pattern $x'_i$, which can be any number between one and the number of investigated classes $C$.

**Confusion matrix** After classifying the patterns $x'_i$ via $\Upsilon$, one has to measure the success of the classification. Hereafter, all measures of performance relate to the confusion-matrix $\Gamma \in \mathbb{N}^{C \times C}$ defined below\(^8\):

$$\Gamma_{ij} = \sum_{m=1}^{N} \delta_{i,\Upsilon(x_m)} \delta_{j,\hat{x}_m} = \sum_{x_m \in \hat{x}_j} \delta_{i,\Upsilon(x_m)} \quad (2.13)$$

$$\Gamma_{ij} = \text{# of vectors } x \text{ that are elements of the } j^{th} \text{ class and classified as class } i \quad (2.14)$$

In eq. 2.13, $\hat{x}_j$ is the set comprising all feature vectors of class $j$.\(^9\) Moreover, $N$ is the number of heartbeats (patterns) in the data under test. The number of heartbeats comprising the $j^{th}$ class, denoted as $N_j$, is simply the cardinality of $\hat{x}_j$, i.e. $N_j = |\hat{x}_j|$. One easily checks that $N = \sum_j N_j$ holds. With those definitions one can easily derive the equations for overall classification accuracy $ACC$ and sensitivity $T_j$ respectively specificity $S_j$ for the $j^{th}$ class. The accuracy can be measured as:

$$ACC = 1. - \frac{1}{N} \sum_{i=1}^{C} \sum_{j \neq i} \Gamma_{ij} \quad (2.15)$$

The sensitivity $T_j$ respectively the specificity $S_j$ for the $j^{th}$ class are defined as:

---

\(^7\) The normalization coefficients are calculated on the training set only. \(^8\) Defining $\Upsilon$ as a surjection ensures that the classifier is not indecisive. Therefore, for each pattern there exists a unique map into $\{1, \ldots , C\}$. Therefore, the confusion matrix $\Gamma$ ignores the possibility of classifier indecision. \(^9\) For the $C = 13$-classification task at hand, $\hat{x}_1$ is the set of all feature vectors that were annotated as normal beats, $\hat{x}_2$ is the set of all paced beats and $\Upsilon$ is the multi-class support vector machine algorithm.
2.3 Mathematical framework

\[ T_j = \frac{TP_j}{TP_j + FN_j} = \frac{\Gamma_{jj}}{N_j}, \quad S_j = \frac{TN_j}{TN_j + FP_j} = 1 - \frac{\sum_{i \neq j} \Gamma_{ij}}{\sum_{i=1}^{C} \Gamma_{ij}} \]  

(2.16)

**Alphabetical short summary**

The table below concludes the latter definitions in a brief summary:

<table>
<thead>
<tr>
<th>symbol</th>
<th>object description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_i )</td>
<td>the ( i )th sample of the raw ECG-signal</td>
</tr>
<tr>
<td>( t_i )</td>
<td>the ( i )th time step</td>
</tr>
<tr>
<td>( N_{raw} )</td>
<td>the number of recorded samples in the ECG</td>
</tr>
<tr>
<td>( \tau_s )</td>
<td>the sampling interval, i.e. the distance between each measurement</td>
</tr>
<tr>
<td>( f_s )</td>
<td>the sampling frequency of the ECG</td>
</tr>
<tr>
<td>( R_i )</td>
<td>the sample number of the ( i )th beat in the raw ECG-signal</td>
</tr>
<tr>
<td>( Y(\Delta_{seg}) )</td>
<td>the raw beat matrix for all ( N ) beats containing the ( \Delta_{seg}/2 ) samples before and the ( \Delta_{seg}/2 ) samples after the R-peak denoted by ( R_i )</td>
</tr>
<tr>
<td>( N_b )</td>
<td>the number of beats in the raw signal</td>
</tr>
<tr>
<td>( \iota_j )</td>
<td>the indices of the heartbeat comprising the ( j )th class</td>
</tr>
<tr>
<td>( Y'(\Delta_{seg}) )</td>
<td>the matrix containing the preprocessed beat segments</td>
</tr>
<tr>
<td>( y'_i )</td>
<td>the ( i )th row-vector of ( Y' ) containing the segment-samples of the ( i )th beat</td>
</tr>
<tr>
<td>( \phi )</td>
<td>a feature generator: transformation of a segment vector ( y'_i ) to a scalar value</td>
</tr>
<tr>
<td>( F )</td>
<td>feature set: a set of several feature generators ( \phi_i )</td>
</tr>
<tr>
<td>( \Phi )</td>
<td>the raw feature matrix generated from the preprocessed data using the set of feature generators ( F )</td>
</tr>
<tr>
<td>( f_i )</td>
<td>the ( i )th feature vector, i.e. the ( i )th column vector of ( \Phi )</td>
</tr>
<tr>
<td>( x_i )</td>
<td>the ( i )th raw pattern(^{10} ), i.e. the ( i )th row vector of ( \Phi )</td>
</tr>
<tr>
<td>( \Xi(EFF) )</td>
<td>the interclass scoring table for a certain effect function</td>
</tr>
<tr>
<td>( EFF )</td>
<td>an effect function measuring discriminancy between two vectors</td>
</tr>
<tr>
<td>( f'_i, f^i, f )</td>
<td>the class dependent feature vectors for the ( i )th feature of the ( j )th class</td>
</tr>
<tr>
<td>( C )</td>
<td>number of classes that will be analyzed</td>
</tr>
<tr>
<td>( \Phi' )</td>
<td>the normalized feature matrix</td>
</tr>
<tr>
<td>( f'_i )</td>
<td>the ( i )th normalized feature vector, i.e the ( i )th column vector of ( \Phi' )</td>
</tr>
<tr>
<td>( x'_i )</td>
<td>the ( i )th pattern(^{11} ) with normalized features, i.e. the ( i )th row vector of ( \Phi' )</td>
</tr>
<tr>
<td>( N )</td>
<td>the number of overall patterns to classify, i.e. the overall number of beats</td>
</tr>
<tr>
<td>( \Upsilon )</td>
<td>a general classifier, e.g. support-vector machines neuronal networks etc.</td>
</tr>
</tbody>
</table>

---

\(^{10}\) ECG-beat \(^{11}\) ECG-beat
Table 2.5 – Continued from previous page

<table>
<thead>
<tr>
<th>symbol</th>
<th>object description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma )</td>
<td>the ( C \times C ) confusion matrix</td>
</tr>
<tr>
<td>( \tilde{X}_i )</td>
<td>the set of pattern vectors ( x'_i ) that comprise the ( i^{th} ) class</td>
</tr>
<tr>
<td>( N_j )</td>
<td>the number of patterns(beats) in the ( j^{th} ) class ( N_j =</td>
</tr>
<tr>
<td>ACC</td>
<td>the accuracy of a classification</td>
</tr>
<tr>
<td>( T_j )</td>
<td>the sensitivity of the ( j^{th} ) class</td>
</tr>
<tr>
<td>( S_j )</td>
<td>the specificity of the ( j^{th} ) class</td>
</tr>
</tbody>
</table>

Table 2.5: Summary of the most important mathematical objects used in this thesis
2.4 The $\rho$-Score: a framework for comparing ECG classifiers

Clearly defining the mathematical framework in which the classifier is built is not sufficient. One also needs to define a framework in which reliability and performance of the classifier can be judged. In the last decades, dozens of ECG-classifiers were proposed in literature, yet there is an urge to compare those classifiers with one another. This is quite complicated, because the proposals are based on quite different setups by means of data composition, amount of arrhythmias investigated and usage of training and test data and so forth.

This section will introduce the Reliability Score ($\rho$-Score). It enables the qualitative and quantitative assessment of the reliability of the classification approaches in the literature. The result-section 6.1. demonstrates that there is an anti-correlation between the reliability of a classifier and its performance measured by the $\kappa$-index which was introduced by Mar et al. \[7\] and generalized for an arbitrary amount of classes $C$ in here.

2.4.1 Criteria of reliability

This review intends to give the reader a hint which proposal is most likely to be a good starting point for further research and improvement of ECG-classifiers for real-life applications. Many great ideas have been tested using only a small amount of data showing great results. On the other hand, less sophisticated approaches have been tested on a wide data range. Some authors communicate only basic performance statistics, whereas others provide all information needed to derive all reasonable performance measures. This great variety of documented and undocumented information about the proposed ECG-classifiers reduces comparability. That is why the current frontier of research remains foggy. This review is intended to dissipate this fog.

The criteria for this review are carefully chosen, but researchers who are more focused on classifier complexity, for instance, would eventually come up with other criteria or criteria-weights. The basic idea is to reduce the impairment of comparability due to incomparable underlying datasets, different optimization techniques and targeted success measures (accuracy, weighted accuracy, mean sensitivity etc.).

Thus, the reliability-score ($\rho$-Score) will assess the following measures:

- Usage of standard ECG-database
- Amount of data
- Ratio between training and overall data
- Selection of data
- Unknown Beat included
- Amount of features
- Confusion-matrix
• Cross-validation with other database

Usage of standard ECG-database  The usage of a standard ECG-database is one of the most important criteria for generalization capabilities of the results, because it guarantees that the confusion matrix was generated on widely accepted annotated signals that can be used as a gold-standard for classifier verification. A classifier verified on either the AHA, MIT-BIH Arrhythmia or CSE database will score 3 points, whereas proprietary data scores only one or zero points (see table below).

<table>
<thead>
<tr>
<th>Database used for training/testing</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHA, CSE, MIT database used</td>
<td>3</td>
</tr>
<tr>
<td>proprietary (private) data, supervised</td>
<td>1</td>
</tr>
<tr>
<td>proprietary (private) data, unsupervised</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.6: Scoring for ECG-database usage

Even though the above criterion is sound, the mere approach of developing a perfect classifier on a large ECG-database (in-house or available to the public) and ship this static classifier is argued by some researchers. Hu et al. [30] emphasized three main pitfalls of this strategy, which are quoted below:

1. "No matter how large this database may be, it is not possible to cover every ECG-waveform of all potential patients. Hence, its performance is inherently limited.

2. The complexity of the classifier grows as the size of the training database grows. When a classifier is designed to correctly classify the ECG from millions of patients (if it ever becomes possible), it has to take numerous exceptions into account. The result is a complicated classifier which is costly to develop, maintain, and update.

3. It is practically impossible to make the classifier learn to correct errors during normal clinical use. Thus, it may be rendered useless if it fails to recognize a specific type of ECG-beats which occurs frequently in certain patient’s ECG-records.”[30]

Hu et al. [30] concludes that algorithms should adapt for the patient at hand. Or, as Watrous and Towell [31] states: “One of the factors that makes ECG-patient monitoring difficult is the variability in morphology and timing across patients, and within patients, of normal and ventricular beats. This variability can create problems for classification algorithms which attempt to classify beats as normal or abnormal on the basis of absolute, patient independent criteria.” Patient-dependent classifiers can be created either by introducing local experts (MOE: mixture of experts) trained on annotated signals of the physician[30] or by normalizing features to the patient at hand[31].

For the ECG machine vendor and the physician in the field, patient adaption poses several disadvantages:

• "While it is possible to turn over training algorithms and databases to the users in an academic environment, it is unlikely that any commercial ECG machine vendor is willing to risk revealing their proprietary information to their competitors. Moreover, in-house database often contains millions of ECG records which could be costly to distribute.

• Users often do not want to be bothered by implementation details of an ECG algorithm.

12 also used by [32]
Thus, few users will be able to take advantage of this patient-adaptation feature even if it is available.

- Even if a user is willing to perform the patient customization, he or she still have to provide sufficient number of patient-specific training data in order to perform patient-adaption. Manually editing ECG record is a time consuming, labor intensive task. Hence, the size of patient-specific training data must be tightly controlled.\"[30]

Yet, there is no solution for the quandary between labor-intense signal annotation and global non-adapting expert systems. New technologies like broad-band Internet, cloud-computing and cheap computing resources may conceivably be part of a potential solution. Nowadays, classifiers that can be updated via Internet taking into account the specific ECG at hand are surely feasible and could improve reliability of the decisions made by the computer program. Moreover, misclassified ECG-snippets could be easily sent to the classifiers’ vendor to adapt its classifier to the new data and then send an update for the classifier to all clients.

**Amount of data** The amount of data\textsuperscript{13} that was used in the training and testing process is crucial for understanding the general applicability of a certain classifier. Therefore, the amount of data will be scored according to tab. 2.7:

<table>
<thead>
<tr>
<th>amount of data $r_d$</th>
<th>≤5%</th>
<th>&gt;5%</th>
<th>&gt;10%</th>
<th>&gt;25%</th>
<th>&gt;50%</th>
<th>&gt;90%</th>
<th>=100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2.7: Scoring for amount of data

The amount of data $r_d$ is calculated with regard to the beat types the classifier tries to analyze. Therefore, $r_d$ is calculated as the sum of all used training and test beats normalized by the number of available beats of those types in the whole database. In case a classifier uses 10% of the data, it scores -1 points; a classifier using 95% would score 3 of 4 points. This is not a “nice to have” requisite. Studies by [35] showed tremendous changes in performance when facing low data amounts (4 records: sensitivity 98.36\% for VPCs) and high data amount (all records: sensitivity 67.60\%). This is also supported by findings of Inan et al. [36] showing a performance deterioration when comparing classification results on 7 and on 40 records of the MIT-BIH Arrhythmia database.

**Ratio between training and overall data** The general applicability of the classifier is not only affected by the overall amount of data. The fraction of training data $r_t$ relative to the overall data, represents the portion of training data\textsuperscript{14} is necessary to get a certain accuracy on the test set. So, if one uses 99 patterns for training and subsequently tests the classifier on a single test pattern ($r_t = 0.99$), the generalization ability of the classifier is doubtable, even though it could probably be a perfect classifier.

<table>
<thead>
<tr>
<th>$r_t$</th>
<th>&gt;90%</th>
<th>&gt;75%</th>
<th>&gt;50%</th>
<th>&gt;25%</th>
<th>&gt;10%</th>
<th>&gt;5%</th>
<th>≤5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2.8: Scoring the ratio between training and overall data

\textsuperscript{13} [33] use the number of investigated beats for comparison between different QRS-detectors. The same is true for [34] comparing QRS.classifiers by investigated beats and accuracy.\textsuperscript{14} clearly the validation set for classifier optimization accounts for the training data.\textsuperscript{15} [37] provide some examples for their results at $r_t=25\%$, $r_t=50\%$ and $r_t=75\%$, they partially show tremendous sensitivity ranges (89.8\% to 96.2\%) or neglectable results (95.8 to 97.4\%).

25
Selection of data  Even in case of high \( r_t \) and \( r_d \), the process of selecting the data points (records and beats) is a big issue. The most valid approach would be to randomly select a portion of the complete database at hand as the training set and use a disjoint set for testing.

Discarding beats for non-technical reasons will artificially decrease intra-class variance. On the other hand, focusing on the idiosyncrasies of each class only will yield poor outcome in a real-life application. Therefore, supervised record or beat selection should be avoided. Approaches using a supervised scheme will be penalized with minus one \( \rho \)-Score.

The dependence or correlation of the ECG-beats is a major loophole for introducing optimistic bias to the measures performance. As the ECG-records originate from only a limited number of patients, the measured heartbeats are not independent. Therefore, splitting the ECG-database on a beat and not on a patient basis is somehow problematic. de Chazal et al. [38] could substantiate, “that dividing data on a beat basis results in optimistically biased classification results. This scheme is biased as intrasubject beats can be highly dependent and hence dependent beats may appear in the training and testing sets.”[38]

Therefore, the most appropriate selection scheme is the use of several randomized disjoint and independent test and training sets.

An approach for independent test and training sets was proposed by de Chazal et al. [38], which gained near-gold-standard status\(^16\) in case of AAMI accordant ECG-classification. The AAMI recommends\(^[39]\) using only five ECG-types for classification\(^17\): normal, atrial premature, ventricular premature, fusion and unclassified beats.\(^18\) The independence of the datasets is ensured by splitting the records of the MIT-BIH Arrhythmia database into two sets of 22 records\(^[19,20]\), the DS1 and DS2 set. Hence, the DS1 set will not contain dependent beats, in case the optimization is performed on the DS2 set. It is apparent that diligently choosing DS1 and DS2 to guarantee equal distribution of classes circumvents the interchangeability of records between DS1 and DS2.

This is a major drawback of this approach, because one can’t measure the dependence of the test performance on this very special choice of record composition. With the currently available databases, one is left with one of two imperfect choices.

Either one has a bootstrapped performance evaluation of dependent data or a single outcome estimate\(^21\) for independent data. Randomized disjoint datasets using independent data seems out of range. Therefore, both selection strategies will be rated equally.\(^22\)

<table>
<thead>
<tr>
<th>Data selection method</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>supervised selection of datapoints</td>
<td>-1</td>
</tr>
<tr>
<td>unknown</td>
<td>0</td>
</tr>
<tr>
<td>random disjoint datasets, DS1/DS2 split</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2.9: Scoring for data selection methods

Unknown Beat included  As mentioned in the previous paragraph using unknown beats for classification is still subject to debate. To the author using an unknown beat is obligatory. In some cases one could also summarize all beat-types of minor interest into a single beat type: the OTHER-beat. Even

\(^{16}\) used by[7]  \(^{17}\) records containing paced beats are ignored  \(^{18}\) some authors\([20, 40–42]\) refuse using unknown beats due to their poor representation in the dataset(see [7])  \(^{19}\) records using paced beats were excluded, which is recommended by AAMI[39].  \(^{20}\) Both sets have nearly equal class distributions.  \(^{21}\) no bias or variance estimates can be retrieved  \(^{22}\) Works by Burman et al. [43] suggest that independence might be accomplished in case the ECG-beats used for training are only a sparse selection of the surrounding beats in the record.
though some beat-types maybe poorly represented in the database, the agglomeration of all beats that are not targeted by the classifier, i.e. the OTHER-beat, is often well represented. Obviously, in case there is a beat that is unknown to the classifier, which is likely for real-life usage, the classifier should tell, that it is an unusual beat and not arbitrarily opt for any beat it is trained for. Moreover, it is favorable to have an unknown beat, in cases of high noise, signal failure etc.. In these situations, a classifier could potentially classify this as unknown, so the physician can take a closer look at that certain beat.

Despite of those advantages, the specificity becomes more valid. Specificity measures how many beats that are not of class A will be classified as not of class A. So by testing more beats that are not of class A, but of the OTHER class, the specificity measure gets more accurate. Moreover, widening the definitions of the beats from 6 to 15 beats improves the quality of the algorithm, because a priori it is unknown if left bundle branch blocks for example are close to normal beats within the high dimensional feature space. If not, the classifier will have a hard time discriminating between those classes.

In conclusion, using an unknown beat or OTHER-beat class in classification adds two $\rho$-Score points to the overall assessment.

**Amount of features** Another criterion to measure the generalization power of the classifier is the amount of features it employs. The less information(features) needed to give a high classification power, the more likely it is that those features are a general and concise description of the patterns under investigation. Because many classifiers differ in the number of ECG-subclasses that they can classify, one should normalize the number of features to the number of classes according to eq. 2.17.

\[
p_c = \frac{\text{# of features}}{\text{# of subclasses}} = \frac{F}{C} \tag{2.17}
\]

The table below rates the normalized amount of features:

<table>
<thead>
<tr>
<th>Amount of Features $p_c$</th>
<th>$&lt; 1$</th>
<th>$&lt; 2$</th>
<th>$&lt; 4$</th>
<th>$&lt; 7$</th>
<th>$&lt; 10$</th>
<th>$&lt; 13$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
</tbody>
</table>

Table 2.10: Scoring for number of features

**Measures of Performance** The criteria above focus on the data the classifier is based on. Another important issue is how the performance of the classification is measured and for which measure it is optimized for. Depending on the author’s intentions for the application of the ECG-classifier, different measures of performance can be employed. To understand how those measures describe different aspects of the classification power of the algorithm, one has to understand how they relate to the confusion matrix from which they are calculated (see mathematical definitions).

Measuring the performance by accuracy or by the mean per class sensitivity can lead to tremendous differences in the interpretation of the classification performance. This is due to the fact that the accuracy is not normalized to the number of elements $N_j = |\tilde{X}_j|$, whereas the mean sensitivity is. This can be easily understood with regard to the following example:

23 [40] shows a dramatic decrease in performance when faced with a new class (the fusion beat), which is a plausible real-life scenario.

24 $T = \frac{1}{C} \sum T_i$
\[
\Gamma = \begin{pmatrix}
90 & 90 \\
10 & 510
\end{pmatrix} 
\]  
\[\text{acc} = \frac{90 + 10}{100 + 600} = 86\%\]  
\[T = \frac{1}{2} \left( \frac{90}{100} + \frac{510}{600} \right) = 88\%\]

In a two-class classification task with \(N_1 = 100\) and \(N_2 = 600\) the algorithm gives the above confusion matrix \(\Gamma\) with accuracy 86\% and mean sensitivity. If the algorithm for classification is trained to optimize the accuracy only, one could also obtain the following confusion matrix:

\[
\Gamma = \begin{pmatrix}
50 & 50 \\
50 & 550
\end{pmatrix} 
\]  
\[\text{acc} = \frac{50 + 50}{100 + 600} = 86\%\]  
\[T = \frac{1}{2} \left( \frac{50}{100} + \frac{550}{600} \right) = 71\%\]

In the example above, the accuracy is just the same as in the previous confusion matrix, but the mean sensitivity has decreased by 17 percent points. Clearly, it is the choice of the classifier's designer to choose the optimization strategy (accuracy, mean sensitivity etc.). Nevertheless, the results should be presented as a confusion matrix so that a practitioner can decide which classifier suits their purpose (high accuracy, high sensitivity or something else). Moreover, the confusion matrix should be accompanied by error measures calculated by bootstrapping or jackknifing to give an estimate about the bias and variance of the result.

<table>
<thead>
<tr>
<th>Performance Measures provided</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>no confusion matrix</td>
<td>-5</td>
</tr>
<tr>
<td>accuracy and sensitivities</td>
<td>1</td>
</tr>
<tr>
<td>confusion matrix</td>
<td>3</td>
</tr>
<tr>
<td>confusion matrix reconstructable (rec.)</td>
<td>3</td>
</tr>
<tr>
<td>confusion matrix with errors</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2.11: Scoring for confusion matrix

**Cross-validation with other database**  The cross-validation of a certain classifier on an other database is the final challenge for any ECG-classifier. A generalizable classifier should be robust against slight changes in electrode-positioning, independent from the type of ECG-recording device and different levels of noise in the data. Therefore, results of an algorithm trained on one database and cross-validated on another database will provide valid information about the generalization ability of a certain algorithm.

So finally, a classifier can score 30 points in the validity-generalization assessment.

\(^{25}\) In a two-class scenario the mean per class sensitivity is equal to the balance rate and the mean per class specificity \(^{26}\) Consumes a lot of computer resources
2.4 The $\rho$-Score: a framework for comparing ECG classifiers

### Table 2.12: Scoring cross-validation with other database

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Scoring example  Now let us take a closer look at a scoring example. The table below is one of 65 tables showing the scoring results of different ECG-classifiers which can be found in the appendix of this thesis.

As can be seen from the table below, Prasad and Sahambi [17] use the standard MIT-BIH Arrhythmia database scoring 3 of 3 points. Moreover, they use nearly the whole database (99%) and use 28.8% of the dataset as training data. So they score 3 of 4 points for data amount and 3 of 8 for the training set ratio. As they try to discriminate between 13 different class, they can use 25 features and still score 2 of 6 points. Unfortunately, they did not present the whole confusion matrix, but one could easily deduce accuracy and mean-class-sensitivity so they score 2 of 5 points. The final score is 13 of 30 possible points, which is quite a high score even without using a cross-validation database.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>Prasad and Sahambi [17]</td>
<td></td>
</tr>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_d$</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_{rt}$</td>
<td>28.8%$^a$</td>
<td>3/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_f$</td>
<td>$\frac{23}{17} = 1.9^b$</td>
<td>4/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>acc, $T$</td>
<td>1/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>17/32</td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>13 $^c$</td>
<td></td>
</tr>
<tr>
<td>acc</td>
<td>96.77%</td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>86.62%</td>
<td></td>
</tr>
<tr>
<td>$S$</td>
<td>$-$-%</td>
<td></td>
</tr>
<tr>
<td>$P$</td>
<td>$-$-%</td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>$-$-%</td>
<td></td>
</tr>
<tr>
<td>$\kappa$</td>
<td>$-$-%</td>
<td></td>
</tr>
</tbody>
</table>

$^a$ some beats with 25%, some with 50%, calculated total train-set/total test-set  
$^b$ 23 DWT coefficients, 2 RR-intervals  
$^c$ N, LBBB, RBBB, AAB, VPC, FVN, NPREM, APC, VESC, NESC, vP, VE, FPN

### 2.4.2 Measuring performance: The $\kappa$-index

In the last subsection, several criteria have been discussed in order to rate the reliability of the classifier. Moreover, it was demonstrated in the paragraph about performance measures that it is quite hard to break down classification results into a single performance score. Mar et al. [7] emphasizes the point that the accuracy (or multi-way accuracy, as they call it) is dominated by the number of true positive normal beats and therefore, "if all beats were labeled as normal, multi-way accuracy would still retain a high value of over 89%." This fact necessitates the weighting of accuracy by the number of beats each class under investigation contributes to the test set. Therefore, Mar et al. [7] introduce the $\kappa$ measure, which "globally evaluates the confusion matrix[44]"[7].
Even though $\kappa$ accounts for the high-skewed distribution of samples per class, it does not evaluate the values of off-diagonal elements in the confusion matrix. For this purpose, Mar et al. [7] introduce the $j$-index, which is simply the sum of predictive values and sensitivities of the V and S beats. As the subsequent review of classifiers incorporates many different class type setups, the $j$-index will be generalized according to:

$$j-\text{index} = \frac{1}{2 \cdot C} \sum_{j=1}^{C} T_j + P_j$$

(2.25)

In the final review the average of the $j$-index and $\kappa$, the so called $j\kappa$-index, is used to rate the performance.
Chapter 3
Preprocessing

The end of the last chapter showed that the raw ECG-data contains lots of noise and inter- and intra-patient variability. That is why preprocessing plays a major role in the whole classification process and is a potential pitfall for all further analysis. The preprocessing stage should assure that information necessary for classification is not altered, hidden or obscured by noise, artifacts or other signal impurities. The challenge is to eliminate as much corrupting information as possible while preserving the majority of the signals characteristics. This chapter describes the steps necessary in order to find the QRS-complex\(^1\), eliminate "powerline interference, high-frequency electromyographic noise and low-frequency drift"\(^2\), and prepare the signal for the subsequent feature extraction stage. This whole process is founded on the three pillars of ECG-preprocessing\(^3\): the removal of high frequency noise, the elimination of power-line interference and baseline estimation for baseline wander removal.

The technologies for noise removal are vast. This thesis will stick to standard preprocessing, according to the AHA's recommendations, and best current practice in QRS-detection and -delineation. This is for two reasons: First, the major technologies to increase preprocessing quality use neural networks and other supervised learning techniques. This poses the risk of over-fitting in a quite early stage of the classification process. Secondly, the focus of this thesis is the investigation of feature extraction techniques and proper feature selection. Hence, using preprocessing optimized for the data at hand will favor less robust features. Those features are only viable for high quality signals. In more noisy environments, which the optimization does not account for, they are potentially weaker. The best practice is not to filter all noise contributions, as their frequency spectrum overlaps with the spectrum of the ECG. It is better to incorporate a noise alarm for the QRS-detector.\(^4\)

The outline of this chapter is as follows: First, the characteristics and principles of FIR-filters are explained. This will enable the reader to understand basic problems of the FIR-filter approach. Furthermore, it eases the understanding of the connection between filter banks and wavelet transformation, discussed in section 3.3. The second subsection will focus on the different noise contributions, such as noise originating from the body or external influences that affect the ECG-recorder itself. The subsection concludes with the description of the final setup. It also shows the Bode-diagram of the final filter. (see \(^4\) for similar approaches) The third part of the chapter gives a brief summary about the QRS-detection via the Pan-Tompkins algorithm and the QRS-delineation via ECGpuwave. The chapter will conclude with the normalization strategies used to decrease inter- and intra-patient variability of the ECG. It also describes the ECG-segmentation strategy.

\(^1\) respectively the fiducial point
3.1 FIR-Filter

A Finite-Impulse-Response (FIR) filter is a linear filter \( H \) that convolves the input signal \( y_n \) with a finite number of filter coefficients \( h_n \):\(^2\)

\[
y'_n = H(y) = \sum_{k \in \mathbb{Z}} h_k y_{n-k}
\] (3.1)

One can easily deduce from eq. 3.1 that FIR-filters are BIBO-stable linear time invariant systems. This means that there is no feedback-loop updating filter coefficients and that the output is bounded in the case that the input is too. The following paragraphs will discuss the four most important characteristics that have to be considered during filter design: the impulse response, the magnitude response, the phase response and the group delay.

The latter three are most important with regard to practical considerations, as they describe:

- how strong certain frequencies are amplified or damped (magnitude response)
- how strong frequency contributions are shifted in time (phase response)
- and by how much a certain structure in the signal (e.g. QRS-complex) is time-shifted (group delay)

However, understanding the impulse response is essential for deriving those three characteristics.

**Impulse response**  The impulse response is the response of the filter on the impulse function:

\[
\delta_n := \begin{cases} 
1 & n = 0 \\
0 & n \neq 0 
\end{cases}
\] (3.2)

The impulse response of any FIR filter simply yields its filter coefficients:

\[
y'_n = H(\delta)_n = \sum_{k \in \mathbb{Z}} h_k \delta_{n-k} = h_n
\] (3.3)

In fact, if \( H(\delta) \) has compact support, the filter has finite impulse response and is, therefore, a FIR-filter. Now by using the z-transform:

\[
H(z) = \sum_{k \in \mathbb{Z}} h_k z^{-k}
\] (3.4)

\[
Y(z) = \sum_{k \in \mathbb{Z}} y_k z^{-k}
\] (3.5)

---

\(^2\) see section “3.1 Digitalfilter” in [49]
the filter equation 3.1 reads as:

\[ Y'(z) = H(z) \cdot Y(z) \] (3.6)

which is simply a multiplication of the z-transforms of the signal and the filter. So, the major question is: How does the transfer function \( H(z) \), which describes the whole filter, act on the signal? How does it change \( y(t) \)?

**The Bode diagram** Clearly, as \( y(t) \in L^2 \), \( y(t) \) can be written using the basis functions \( e^{i2\pi \xi} \), yielding:

\[ y(t) = \langle y | e^{i2\pi \xi} \rangle = \sum_{k \in \mathbb{Z}} h_k e^{i2\pi \xi k} \] (3.7)

In order to understand the influence of the filter \( H(z) \), one simply needs to focus on its influence on the basis functions \( e^{i2\pi \xi} \). Let \( y_n = e^{i2\pi \xi n} \) be the input for the filter \( H \). The filtered output \( y'_n \) is given by:

\[ y'_n = \sum_{k \in \mathbb{Z}} h_k y_{n-k} = \sum_{k \in \mathbb{Z}} h_k e^{i2\pi \xi(n-k)} = e^{i2\pi \xi n} \sum_{k \in \mathbb{Z}} h_k e^{-i2\pi \xi k} = \hat{h}(\xi) y_n \] (3.8)

Now, the filtered signal \( y'(t) \) can be expressed by means of the Fourier transform of the filter coefficients \( \hat{h}(\xi) \):

\[ y'_n = \sum_{k \in \mathbb{Z}} h_k y_{n-k} = \sum_{k \in \mathbb{Z}} h_k \int_{-\infty}^{\infty} \hat{y}(\xi) e^{i2\pi \xi(n-k)} d\xi = \int_{-\infty}^{\infty} \hat{y}(\xi) \hat{h}(\xi) e^{i2\pi \xi n} d\xi \] (3.9)

The resulting Fourier-transform of the filtered signal is simply the multiplication of the signal’s original Fourier-transform and the Fourier-transform of the filter coefficients. For a better visual analysis of the impact of \( H(z) \), one investigates the Bode-plot, i.e. plotting the magnitude response \( |\hat{h}(\xi)| \) and the phase response \( \text{Arg}(\hat{h}(\xi)) \) of the transfer function side by side. The magnitude response tells how strongly a certain frequency \( \xi \) is amplified or damped, whereas the phase response measures the phase shift of a sinusoid of this frequency caused by the filter.

**Group delay** Obviously, the shift in phase will distort the signal because the phase shift is generally speaking non-linear and frequency-dependent. Moreover, the amplitude envelope of any frequency is shifted by a certain amount of time. This delay is measured by the group delay:

\[ \tau_g = -\frac{d \text{Arg}(\hat{h}(\xi))}{d\xi} \] (3.10)

---

3 see [49] 4 This could have also been deduced from eq. 3.6, because clearly \( \hat{y}'(\xi) = Y'(e^{i2\pi \xi}) \)
Clearly, $\tau_g$ can only be constant and, therefore, frequency-independent in the case of a linear phase response.

**Filter design restrictions** What should the filter $H(z)$ look like? The magnitude response of an ideal filter would have the shape of a step function. Thus, it should remove all frequency content up to a certain cut-off frequency while retaining all other frequency content without any alteration. Moreover, this filter should have zero group delay. This holy grail does not exist. The design of the filter is not only a trade-off between group delay (small in the case of little support of $h$) and strong decay at the cut-off frequency (stronger if support is big). One also has to consider Gibbs-phenomenon. The magnitude response will suffer from an overshoot and subsequent ringing at the edges of the targeted step function. This phenomenon is caused any time one tries to express a step function in a basis of polynomials with a finite number of basis functions. In the light of the Gibbs-phenomenon, the magnitude response of the FIR filter has to be tuned in order to have a slightly smooth, but still strong decay in order to match the necessities of signal processing.

The level of group delay, on the other hand, is of minor importance as long as the record duration is significantly higher than the group delay. Consider that each recording in the database is 30 minutes long. Hence, a filter with group delay of approx. 5 seconds (1500 samples) will drop 5 beats per record on average.\(^5\)

### 3.2 Filtering noise and artifacts

The proper preprocessing of the ECG is still a demanding challenge (see Gregg et al. [51] for a comprehensive analysis). “The directional nature of ECG makes it highly variable with respect to the position, size and anatomy of the heart even among normal people. The manner in which the heart’s electric phenomena is led to the skin also plays an important role(1). In general, age, sex, relative body weight, chest configuration, and various other factors make ECG variant among persons with same cardiological conditions (2). Due to these wide variations exhibited by ECG, the diagnosis using this tool is intuitive and highly subjective. Interpretation of ECG-patterns, particularly when noise and patient factors are present becomes very difficult.” [52] Even though the problems mentioned in the above quote are serious questions, ECG-system vendors and researchers strive to reduce the impurities within the ECG. In this section, I describe the five main drivers of signal noise:\(^6\):

- electromyographic noise
- electrode dislocation
- baseline wander
- power-line interference
- sampling noise

The level of deterioration of the ECG, caused by these kinds of noise, depends heavily on the overlap of the corresponding frequency spectrum with the ECG’s spectrum. Hence, this thesis will use bandpass

\(^5\) More details about filter design based on windows can be found in [50]. \(^6\) see [53] for further insights about the impact of baseline wander and filtering on the power spectrum
3.2 Filtering noise and artifacts

and notch FIR filtering to account for the noise contributions described later on. Even though this is the dominating strategy, some experts do not concur. Hu et al. [54] emphasize "that the background noise process is time varying. The optimal pass band frequencies changes with respect to time. Hence, some data adaptive method will be desirable." [54] Many methods for adaptive noise removal use multilayer perceptrons [54] or other neural networks [55].

Employing more sophisticated preprocessing strategies is a double-edged sword: On one hand, more elaborate methods increase the quality of the extracted features. On the other hand, neural networks pose the risk of over-fitting in a quite early stage of the classification process. Moreover, features only applicable on high quality signals are potentially not robust in more noisy environments. Furthermore, this thesis investigates the power of feature extraction techniques. Thus, the power of features can be artificially increased by strong preprocessing and vice versa. A standard preprocessing unit without data adaption will increase comparability of features on a solid basis.

Electromyographic noise Let us take a look at the different noise types the signal collects during its way from the heart to the recording device. Let us recall the dipole theory of the ECG. It states that the wavefront of the propagating action potential forms a dipole. The far-field of this dipole is measured by the ECG. It is obvious that the ECG does not only measure action potentials (APs) originating from the heart. The electrode measures the superposition of all action potentials in its vicinity. That is why APs emitted by muscular activity are pollute the ECG. Even worse: the power spectrum of electromyographic noise intersects with the ECG's. Therefore, it "is important for clinical application to know that the minimum recommended cutoff frequency is 100 Hz for adults and 150 Hz for pediatric patients [8 in [51]] and that the 40 Hz low-pass filter should only be used if absolutely necessary." [51] Filtering below 40 Hz will distort the original PQRST-complex significantly, which is unfavorable for the detection of ST-segment elevation. An earlier analysis by de Pinto [56] specifically focused on muscle artifacts joins in: "low-pass filtering is somewhat effective in reducing muscle artifact, but has the disadvantage of reducing the QRS-amplitude, due to the overlapping spectra of the muscle artifact and the "true ECG" "[56].

Following the remarks above, the "AHA" recommends using 100 Hz (see [37, 57–62]) as cutoff frequency. Nonetheless, this cutoff depends on the intended application. Some authors, focused on discriminating between normal and ectopic heartbeats, use 35 Hz as their cutoff [16, 63–66]. They accept the distortion of the ST-segment, which is not that important in this classification task, while retaining higher signal quality at the segments of interest. More generally: "effectiveness of bandpass filtering for noise rejection is limited, however, since noise is not completely separable from the signal in the frequency domain" [46]

The filtering itself is typically performed by a bandpass filter with its higher cutoff frequency at 100 Hz. Even though FIR filters are prevailing, other authors use Butterworth-IIR filters [67], moving average low-pass filtering [16, 63–66] or second phase-corrected moving average [46]. Zadeh et al. [68] use stationary wavelet transforms (see section 4.4.5) for denoising, which is equivalent to bandpass filtering between 0.1 Hz and 100 Hz.

Electrode dislocation/misplacement The best way to reduce electromyographic noise is to position electrodes near or above bones and tissue with low muscular content (rips, below arm). Another constraint, especially in long-term recordings, is that the location of the electrode increases compliance. If its position hinders patient-movements or allows easy disconnects, measurements fail. Thus, some ECG-recorders incorporate a leads-off warning system. (see [51]) It simply uses low-amplitude high-frequency (30 kHz) pulses to identify disconnected leads.
Baseline wander  Clearly, electrodes are not always fully disconnected. Baseline wander, i.e. slight non-linear changes of the ECG’s baseline voltage, are "mainly caused by too high impedance at the electrode interface and by breathing artifacts (typically 15-20 times per min). These have quite low frequency components, typically less than 1 Hz."[62]

A computationally efficient way to get rid of those frequencies is a linear-phase low-pass filter[47, 69]. In contrast to the AHA recommendation of 1 Hz cutoff frequency, other authors prefer 0.1 Hz [37, 58, 59]. Reducing low frequency components directly in the spectrum is quite problematic. This approach assumes that the baseline wander, even though it happens on longer scales, has no contributions to higher frequencies. This is a radical assumption if one keeps in mind that the Fourier transform of a linear function has contributions in all frequency components.

Therefore, many authors employ time domain filtering of the baseline wander. All those methods proceed alike: First, they estimate the ECG’s baseline and subsequently remove it. The resulting signal should not contain baseline wander. The methods for the baseline estimation are vast: cubic spline estimation [59], using the simply the ECG’s mean value [8, 30], linear detrending [67, 70] or sliding-median filters [7, 41, 71] are common.

Even though linear-phase FIR filters and time domain filters are predominant, more sophisticated filter setups were considered in the literature: [16, 63–66] use high-pass recursive filtering for drift suppression with a cut-off frequency of 2.2 Hz. Besides the different filtering approaches, there is no consensus about the cut-off frequency of the high-pass filters. Higher values as proposed by [16, 63, 64, 66] (2.2 Hz) are used as well as lower ones: 0.1 Hz [37, 58–60], 1 Hz [57, 61, 62] and 0.5 Hz [33, 48, 72]. Even lower cutoffs (0.05Hz) are necessary to prevent any ST segment distortion due to high-pass filtering linear phase filters. [51]

For the removal of baseline wander, this thesis will use recommendations in [48] (-0.5db at 0.8Hz) and use a median filter estimate of the baseline in advance to account for non-linear baseline wander contributions. This is in accordance with recommendations of the AHA. [73, 74]

Power-line interference  Despite noise created by processes within the body or at the body-electrode interface, the environment, especially its electro-magnetic(EM) pollution, plays a crucial role. The power-line emits a 60Hz (U.S.) resp. 50Hz (Europe) EM-wave which propagates along the surface of our skin. Hence, it is easily measured using a body electrode.

The removal of this interference seems an easy task. In part, the job is already done by the right leg drive, which "refers to the circuit used by electrocardiographs to reduce common-mode interference. Common-mode interference is an unwanted signal that is common to all electrodes compared to earth ground...".[51] Nonetheless, achieving high signal quality necessitates digital filtering of the signal.

Even though some authors do not account for power-line interference [41, 52, 59, 68, 70, 71], the dominating strategy is to use FIR-notch filters at 60 Hz [57, 58, 61, 62]. Approaches employing moving averages to cancel 60 Hz contributions [16, 63–66] are considered FIR approaches because they are simply a low-pass FIR filter. [12]

---

7 Hargittai [69] uses subsequent downsampling to increase computational efficiency  
8 potentially introduces more inter- and intra-patient variability than it potentially reduces  
9 the drift suppression is described in references [14, 15] in [63]  
10 see [47] for a similar strategy  
11 Besides power-line interference, common-mode interference entails interference caused by differing impedances of the skin-electrodes at different positions on the body. Therefore, feedback systems are used to cancel out those effects.  
12 A promising adaptive non-linear filtering method (time-warped polynomials) [75] seems superior from the graphs they provide. As no hard facts about signal quality improvements are given, this method is not employed in the thesis
Sampling noise  In the last paragraphs, all noise contributions have been considered analog. The last step from the noise prone analog signal into the recorder is the digitization. This task also introduces noise to the signal. The main constraint here is the sampling frequency of the analog/digital converter. "A typical voltage resolution, also the value recommended by the International Electrotechnical Commission performance standards for electrocardiographs, is 5 µV.[51] The discretization of the signal reduces smoothness of the analog signal and, therefore, adds noise to the signal.

Nonetheless, sampling is a priori not evil. The time axis is also sampled. It is even oversampled two or three times. This assures that all frequency components of the ECG can be incorporated. This is advantageous because oversampling allows to remove high frequency components much easier. This is due the prolonging of the transition band: in the case that one wants to remove the frequency content between 60-100 Hz, the transition band is pretty narrow, if the signal is sampled at 200 Hz. Oversampling the signal at 500 Hz, on the other hand, allows the elimination of frequencies in the range from 60 to 250 Hz, which is a much wider transition band and therefore computationally favorable.

Final setup  The major concerns with regard to noise pollution are: baseline wander, power-line interference and electromyographic noise. The non-linear characteristic of baseline wander necessitates the usage of a high-pass moving median filter (180 points). Afterwards, a bandpass FIR filter with a notch at 60 Hz\textsuperscript{13} is employed. The lower cutoff-frequency is 1 Hz, whereas the higher one is 100 Hz.

Even though a recent survey by Narwaria et al. [76] concluded that there is still space for higher accuracies and less distortion of the QRS-complex, this thesis does not strive for perfect preprocessing. Using a non-standard preprocessing stage would heavily weaken assertions about the strength of the extracted features, the feature selection methods and the final classifier. Even though a sophisticated preprocessing could improve accuracy, one would not measure the features’ performance in a standard environment. A preprocessing which is heavily tuned to the signal-to-noise characteristics of the data set at hand, renders comparisons between feature extraction strategies insignificant. Features that are less robust for real-life applications could possibly prove successful, but are useless when confronted with high noise.

The above setup is, therefore, in accordance to the AHA recommendations[73, 74]. It is depicted in the graphs below.

3.3 QRS-complex detection and delineation

The detection of the QRS-complex is the most essential task for the overall classification. QRS-complexes that are not detected can not be classified.\textsuperscript{14} Fortunately, the last decades provide a vast amount of different QRS-detectors proposed in the literature[77]\textsuperscript{15}. The majority of those detectors were tested on the MIT-BIH Arrhythmia database[12]. Due to their high comparability, the author concludes that the accuracy of most of the art QRS-detectors will be sufficient for the classification task at hand.

The author favors employing the ECGpuwave algorithm developed by Laguna et al. [80], which is publicly available and was tested on different standard ECG-databases [80–82]. The procedure is not only highly accurate, it also provides information about the delineation of the ECG. This is quite handy, as the

\textsuperscript{13} The databases used in this thesis are populated with signals measured in the US. For ECG-signals with origin in the EU, especially in Germany, the filter has to use a 50Hz notch filter. [33, 72] use the CSE database and have to account for the European power lines.

\textsuperscript{14} Some authors[7] do use annotation file classification only\textsuperscript{15} A comprehensive review was performed by Köhler et al. [78]; a detailed comparison can be found in [79]

37
Figure 3.1: The magnitude (solid line) and phase (dotted line) response of the final filter used for noise reduction. The lower graph shows the magnification of the high-pass filter with cutoff at 1Hz.
delineation information is needed in the feature extraction stage for some features. Another major advantage is that ECGpuwave is widely accepted and used for ECG-classification purposes [5, 7, 38, 40, 83]. In conclusion, the combination of standard ECG-filtering and QRS-complex detection/delineation allows to measure the performance of features in a typical environment.

3.3 QRS-complex detection and delineation

3.3.1 Pan-Tompkins QRS-detection

The basic, most widely used concept of QRS-detection is thresholding. Simple approaches, like slope [84, 85] or entropy [72] thresholding, exploit the fact that the R-peak causes strong first derivatives which can be easily implemented. Nevertheless, those approaches are not robust with regard to noise, artifacts or steep/big T-waves. Therefore, misclassification has to be reduced. Typically, higher robustness is achieved by bandpass filtering ([86] use a 6-18 Hz bandpass; [87] use a 10-17 Hz bandpass).

The Pan-Tompkins algorithm is more elaborate. It was proposed in [88, 89] and employed by many authors [30, 34, 59, 71, 90–93]. It is, therefore, no surprise that, even though other detectors have been proposed, the ECGpuwave algorithm uses this algorithm as preprocessor. The basic idea is to use a narrow bandpass filter, which removes the noise, and to differentiate and square the resulting signal. In the last step, the squared signal is time averaged using a sliding window of 32 points, which equals 107 ms. The final time averaged signal consists of broad peaks which indicate the position of the heartbeat. The peak is quite trapezoidal: A steep ascent is followed by a plateau and a steep descent. A heartbeat is detected, in the case that the signal reaches half maximum of the plateau within the descent. Marking the fiducial point is performed by searching for the maximum value in the bandpass filtered signal, 250-150 ms before the point of heartbeat-detection. Further decision-rule optimization entails blanking times and search-back strategies. These apply heuristics to assure that detected ECG-beats are reasonable. ECG-beats being to close to each other are mostly related to T-wave over-sensing and ECG-beats too far from each other indicate QRS-under-sensing.

Even though more sophisticated approaches like neural networks [87, 94], support vector machines [33, 72], syntactic methods [95] and linear prediction [96] have been developed, the Pan-Tompkins algorithm is still widely used and popular for its robustness and ease. Nonetheless, it shares a major problem with all bandpass based strategies: "These techniques mainly suffer from two problems: 1) the signal passband of the QRS-complex is different for different subjects and even for different beats of the same subject, 2) the noise and the QRS-complex passbands overlap." [87]

The idea of using bandpass-filtering can be extended by using wavelets, which is, in fact, using multi-level signal analysis employing filter banks. Besides work by [6, 97–99], the wavelet based QRS-detector of [100] is a very powerful one. It is also used by [101, 102] and uses modulus maximum pairs and zero-crossing in different scales of stationary wavelet transform using quadratic splines.16

3.3.2 The ECGpuwave delineation

The algorithms described above focus on the detection of the R-peak resp. the fiducial point of the QRS-complex. More sophisticated approaches try to delineate the QRS-complex, i.e. identifying the P, Q, R, S, T waves as well as their on- and offsets. The determination of those points is essential for further feature extraction. Despite other approaches [93, 96, 102, 104–106] being of potential use, this subsection will focus on the ECGpuwave delineator proposed by [80]. The following paragraphs briefly summarize sections "wave location and wave boundary location" in [80].

[103] use the Mexican hat instead of quadratic spline and enforce LADT.
PQRST points It was demonstrated in the previous subsection that QRS-detection nicely works according to the Pan-Tompkins algorithm. This employs band-pass filtering, differentiation, squaring and averaging. The latter formed an envelope around the QRS-complex indicating the approximate position of the QRS-points. Now, the ECGpuwave software investigates on the bandpass filtered and differentiated signal ECGDER. As the Q, R and S points are maxima resp. minima of the ECG, the algorithm can infer their position by the zero-crossings in ECGDER. Moreover, their polarity and relative amplitude in the filtered signal indicate which peak should be identified with the Q, R or S-wave. After the R-peak is obtained, finding the P and T-wave is straight forward.

The ECGpuwave algorithm accounts for the stronger low frequency contributions of the P- and T-wave by low-pass filtering ECGDER (-3db cutoff frequency at 12Hz). Now the P wave is estimated to be in a window from 380ms to 225ms before the R-peak. The program asserts a P-wave in the case that the maximum and minimum values within the window of the filtered derivative exceed 2% of the QRS-maximum-slope. The P-wave’s position is at the zero-crossing of the window. The T-wave search is performed in the same manner. The major difference is that the window’s position and size is heart rate dependent.

Onset and Offset The determination of the onset and offset is carried out via two different criteria: the "minimum differentiated criterion" [80] and a threshold criterion.

According to the threshold criterion, the QRS-on and -offset are calculated as follows: First, one searches for the next peak in the ECGDER signal, denoted as pk. There, the slope of the filtered ECG (ECGBP) reaches a local maximum. The threshold for onset resp. offset determination depends on the ratio between the local slope maximum and the global slope maximum \( \frac{ECGDER(pk)}{dermax} \). Finally, the threshold is chosen to be \( TH = ECGDER(pk)/k \). k is determined experimentally. Example: Say, the local maximum is up to 40% of the global one. Then, the threshold is one-third \( k = 3 \) of the height of the peak at \( pk \). This procedure is carried out for the P, Q, R, S and T wave. \(^{17}\)

"When two waveforms overlap, the threshold criteria can fail and may give boundary locations far from physiologically expected points." [80] In those situations the "minimum differentiated criterion" is employed. One starts again with the first maximum at \( pk \). However, this criterion will focus on the adjacent minimum, denoted as \( pk' \). It is used as the waveform boundary in cases of ECG-inflection.

The ECGpuwave procedure summarized here can also be tailored to use multi-channel ECG. A major advantage of the described method is that it is mainly based on slope information. Using amplitude thresholding would cause severe problems, as the relative amplitudes in the P, R and T wave are highly variable. They can be altered by slight changes in electrode positioning and orientation of the heart.

\(^{17}\) The table for choosing k can be found in [80]
3.4 ECG-segmentation and normalization

After the QRS-complexes are found the signal has to be split into segments around the R-peak for further classification. There is no standard for choosing the right amount of samples before and after the R-peak. Each author has their own suggestions about a proper choice of the segment surrounding the R-peak. Some authors favor asymmetric segments \([17, 22, 23, 59, 90]\), whereas other's favor symmetric ones \([8, 37, 47, 52, 68, 107–109]\). Moreover, the size of the segment varies a lot, from a segment length below 250 ms \([17, 22, 52]\) or below 500 ms \([37, 47, 59, 68, 107, 109]\) to segment lengths exceeding 500 ms \([8, 23, 108]\) and even above one second \([90]\). The decision for the size of the segment depends heavily on the features that will be used further. A good example can be found in \([110]\), where two window sizes are used: 1000 ms (722 ms before, 278 after) and 556 ms snippets centered around R-peak depending on a threshold RR-interval \(R_R\). Using different window sizes is an obvious asset. Features describing long-term changes and changes in between adjacent R-peaks should be defined on longer segments, rather than segments on which features investigating the QRS-morphology are defined.

Therefore, this thesis will investigate features on three different segments: a pretty short one (410 ms centered around the R-peak), a quite long one (833 ms centered around the R-peak) and one segment that lies in between R-peaks (260 ms before the R-peak to 40 ms before the R-peak). The first segment analyzes the QRS-complex itself, whereas the second also includes the P and the T-wave. The in between segment aims for premature beats which penetrate the normally empty ECG-region (zero-line) between heartbeats of physiologic distance. The segments are computed according to:

\[
Y(\Delta_{seg})_{ij} = y^\prime \left( R_i - \frac{\Delta_{seg} - 1}{2} + j - 1 \right) \cdot \tau_{\text{sampling}} \quad j \in \{1, \ldots, \Delta_{seg}\}
\]

using \(\Delta_{seg} = 150\) (long) and \(\Delta_{seg} = 75\) (short).

**Normalization** After successfully segmenting the signal, one has to normalize these segments, because the raw and even the filtered ECG-amplitudes contain lots of intra and inter-patient variability. This point is stressed by LLamendo and Martinêz \([21]\), who demonstrated that in a cross-database training (MIT-BIH-AR and MIT-BIH-SUP) that their best features \([8]\) "are computed without exception from time interval measurements"\([21]\). The reason is obvious: the databases "do not always include the same pair of ECG-leads in each recording, and naturally the classification performance of features which are calculated from amplitudes are heavily degraded."\([21]\) Besides the well-known problem of ECG-lead selection, several authors proposed normalization strategies which, for example, result "in normalized signals with zero mean and unity standard deviation, which aims to reduce the possible false decisions due to signal amplitude biases resulting from instrumental and human differences."\([10]\) Yu and Chou \([110]\) and Patra et al. \([109]\) subtract the mean value from each segment and divide it by its standard deviation. Others simply normalize the segment between zero and one \([22, 111]\)\(^{18}\) or minus one and

\(^{18}\) the R-wave is normalized to the average R-wave amplitude of the training set
plus one [30]. Some authors normalize for a fixed voltage, e.g. 1mV [8, 98].

As many options are available, this thesis will focus on only three normalizations: MinMax-normalization (MMNorm), Median-removal (MedNorm), No-normalization (NONorm). The latter simply yields the unnormalized ECG-segment. The first normalization strives to align the R-peaks of all ECG-beats on the same level from zero to one. The second normalization is intended to reduce baseline artifacts.

**MinMax-Normalization** At first sight, normalization has many advantages like increasing homogeneity of the data. At second sight, normalization potentially raises a problem in case the QRS-complex does not account for the maximum or minimum in the segment. Large segments can possibly contain other QRS-complexes or artifacts, not filtered by the FIR-filters which could probably distort the segment tremendously due to normalization. In order to reduce these possible drawbacks, this thesis will perform normalization only on a narrow band surrounding the QRS-complex.

Thus the the minimum $m_i$ and the maximum $M_i$ of the $i^{th}$ ECG-beat is only calculated within the assumed QRS-region:

\[
m_i = \min_{k \in \left\{ \frac{\Delta_{seg}}{2} - d, \ldots, \frac{\Delta_{seg}}{2} + d \right\}} (y^f_i)_k
\]
\[
M_i = \max_{k \in \left\{ \frac{\Delta_{seg}}{2} - d, \ldots, \frac{\Delta_{seg}}{2} + d \right\}} (y^f_i)_k
\] (3.12) (3.13)

By using the above definitions, the normalization of the filtered signal reads as (same procedure as in [112]):

\[
\text{MMNorm}: y'_i = \frac{y^f_i - m_i}{M_i - m_i}
\] (3.14)

**Median-Removal** Another source of misalignment and heterogeneity of ECG beats of the same morphology is the baseline. This can be easily corrected by removing the median of the ECG-segment. After median removal one will encounter minima which are more pronounced than the maxima. This QRS-inversion has to be reverted.\(^{19}\)

\[
\text{MedNorm}: y'_i = \begin{cases} 
-m_i & \text{if } |\min(m_i)| > \max(m_i) \\
m_i & \text{else}
\end{cases}
\]
\[
m_i = y'_i - \text{Median}(y'_i)
\] (3.15)

**Summary** The normalized vectors $y'_i$ form the preprocessed beat matrix $Y(\Delta_{seg}, \text{Norm})$. Given the small ($\Delta_{seg} = 150$) and the large ECG-segment ($\Delta_{seg} = 300$), applying all three normalizations yields six different beat matrices $Y(\Delta_{seg}, \text{Norm})$ ready for feature extraction. The in-between segment described earlier yields the seventh beat matrix denoted $Y(\Delta_{seg} = 250')$. The next chapter will describe, in detail, the features that can be extracted from the segments in those matrices.

\(^{19}\) Approaches by [37, 113] are similar, but take not into account QRS-inversion.
Chapter 4

Feature Extraction

The last chapter described the preprocessing of the signal, which ultimately led to the preprocessed and segmented signal that is described by the matrix $Y'$. Each row vector $y'_i$ in $Y'$ represents the segment of a single beat. Its length is $\Delta_{seg}$ and it is centered around the fiducial point of the QRS-complex. This chapter will focus on the vast number of possibilities for extracting features from each vector $y'_i$. Each feature corresponds to a feature-extractor $\phi$ (see Mathematical Definitions chapter for a better overview):

$$\phi : \mathbb{R}^{1 \times \Delta_{seg}} \rightarrow \mathbb{R} \quad (4.1)$$

The aggregate of all feature-extractors, the feature set:

$$\tilde{F} = \{\phi_1, \ldots, \phi_F\} \quad (4.2)$$

is used to calculate the final feature matrix $\Phi$ according to 2.8 (recalled below), which is subject to the final classification.

$$\Phi = Y'(\Delta_{seg}) \odot \tilde{F} \equiv \begin{pmatrix}
\phi_1(y'_1) & \phi_2(y'_1) & \ldots & \phi_F(y'_1) \\
\phi_1(y'_2) & \ddots & \ddots & \phi_F(y'_2) \\
\vdots & & \ddots & \vdots \\
\phi_1(y'_N) & \phi_2(y'_N) & \ldots & \phi_F(y'_N)
\end{pmatrix} = [f_1, \ldots, f_F]$$

Even though there is a rich diversity in feature extraction methods, according to [8, 114], they can be divided into three main branches: direct techniques, transformation techniques and parameter estimation techniques.

**Direct techniques**  "They use the raw amplitudes and directly input it into neuronal networks or other machine learning algorithms, this is pretty easy and comes with low computational cost, but correct R-peak detection and low noise levels are crucial for those algorithms to work."[8]

A major problem in using the raw amplitudes is that the reliability of the features is dependent on the jitter of the R-peak detection. This is a severe threat to clustering techniques or principal component
analysis: As Moody and Mark [46] puts results of [115]: "They emphasize the influence of jitter (sub-optimal fiducial point estimation) on the outcome, and suggest that the discriminating power of the KLT (=PCA) may be enhanced if the training set has jitter comparable to that which will be encountered in the ultimate application." Authors rooting for direct techniques like Hosseini et al. [116] and in partial Mar et al. [7]1 develop low-performing ECG-classifiers. This is due to the excessively high noise on each ECG-sample, which inevitably introduces too much noise to a neuronal network. From a taxonomic point of view, those techniques should not be considered as feature extraction. Introducing the raw amplitudes to the neuronal network is simply not doing anything with the amplitudes. It is like shortcutting the feature extraction and applying classification directly. Clustering techniques and principal component analysis resides mainly in the unsupervised machine learning resp. dimension reduction business. If they are used for feature extraction, they should be considered as transformational strategies which map the signal into a lower dimensional space.

Transformation techniques A study carried out by Acir [8] showed that features from raw data result in poor classification power compared to those originating from Discrete Cosine or Discrete Wavelet Transformation (DCT and DWT). These transformation techniques "Do reduce scattering of feature vectors [114] caused by noise or R-peak maldetection, by transforming the raw signal in Wavelet or cosine domain etc."[8]

Parameter estimation techniques Clearly, the latter techniques are not feature-extractors in the sense of eq. 4.1, i.e. a map of the samples in the ECG-beat's segment to a single real valued number. This is what the parameter estimation techniques are supposed to do. They are "used to find characteristic points P, Q, R, S and T and to use measures like intervals, enclosed areas, amplitudes and so forth to analyze the signal"[8]

This scheme is not very precise, even though this categorization is useful for getting a first glimpse of the possible techniques applicable to the ECG. For example, the autocorrelation of specific segments of the wavelet transformation is a parameter estimation technique employed on a transformation technique. Hence, the category of this feature can not be determined in this scheme. In the author's point of view, parameter estimation techniques are the only real feature extraction techniques. The transformation techniques may or may not improve signal quality or show new aspects of the signal. In the end, after transformation, one is left with a one-dimensional "time series" which is the basis of the parameter estimation. In this respect, the raw amplitudes are simply the result of the analog ECG-amplitudes undergoing a sampling transformation governed by the ECG-device. Therefore, one has to find a proper transformation technique, which is followed by a powerful parameter estimation strategy, to extract the most useful information from the ECG.

Outline The possibilities of feature extraction are vast. For this reason, the first section is devoted to those feature extraction techniques discarded in this thesis and the reasoning behind dropping them. In the second section, the basic principle of parameter estimation in this thesis is explained. From the author's perspective, the majority of features used for ECG-classification follow the scheme: Search a partition of interest in the time series, apply a certain statistic (a partition function) and then compare this statistic with those in other nearby partitions (relational features).

The next sections are all concerned with certain transformations. The third section is concerned with the time-domain and especially the delineation of the QRS-complex. There, the basic differences between the physiologic and static delineation, which was developed in this thesis, are described. The fourth

---

1 use downsampled QRS-complexes and also the T-wave as feature vectors
This thesis will basically use the Discrete Wavelet Transform (DWT) as well as the Stationary Wavelet Transform (SWT). The latter is used due to its translational-invariance and great performance for QRS-detection.

The fifth section explores new possibilities to extract phase information from the ECG. The calculation of the bi-spectrum resp. its equivalent the higher order statistics (HOS) has been used before.\cite{41, 42, 67, 117, 118}. Nonetheless, research remains unsystematic and does not focus on the 2-dimensional structure of the bi-spectrum. This thesis will revive HOS by applying image processing techniques to analyze this structure and extract supposedly powerful features. Although the latter feature-extractors focused solely on one ECG-beat, information about the relation to other ECG-beats might be beneficial. Hence, the sixth section is concerned with RR-intervals and discusses parameters that describe the relation to nearby ECG-beats to increase discriminant power towards VPCs.

This thesis will use around 1.4 million different features. Hence, the last section is intended to keep track of all those features. It summarizes the amount of features used by certain feature extraction strategies. This will ease the understanding of the experiments discussed in chapter 7.

### 4.1 Discarded feature-extractors

Clearly, the feature extraction methods to come are just a selection from a wide spectrum of possible feature-extractors. In the following lines, I would like to refer to some feature extraction techniques in the literature that were not employed in this thesis.

**Independent component analysis** The independent component analysis (ICA) is intended to "express a set of random variables as linear combinations of statistically independent component variables" \cite{110} (for ICA analysis see \cite{119}). Authors like \cite{91, 92, 110, 120} use ICA to classify ECG-beats under the assumption that the sequence of normal and abnormal beats can be described as a superposition of normal and abnormal signals. Supposedly this assertion is erroneous. Assume a discussion between two scientists talking at the same time. ICA is intended to split one hardly understandable discussion into two independent easily digestible talks. Still, this is the exact opposite to the situation in ECG-classification. There one scientist (normal) speaks all the time and is only sometimes interrupted by the other one (abnormal). Accordingly, ICA is a mismatch for ECG-classification. How come some authors show promising results? In the case of the proposal of Yu and Chou \cite{110} they simply transfer the problem back to the two scientists talking gibberish. They intend to distinguish between 6 arrhythmias. Thus, they use 6 patients each suffering from a different arrhythmia. Now, the ICA is trained for distinguishing between those 6 "talks". The final test is also performed with the exact same patients. Consequently, the ICA will identify the patient it was trained for not the arrhythmia. Therefore, this approach will fail on new data. The reader further interested in ICA can refer to a nice introduction given by \cite{120} and the standard book on ICA by Hyvärinen et al. \cite{119}.

**Vector cardiography** Vector cardiography is a new tool which recently gained publicity as a potential risk stratification tool for cardiac arrest. The major concept is to analyze the 3-dimensional propagation vector of electrical activity of the myocardium by combining the integrated two-dimensional information of different ECG-leads. Obviously this approach necessitates two or more leads. Even though some
Chapter 4 Feature Extraction

authors [16, 20, 21, 40, 63–66] make use of vector cardiography, this thesis will not, because most databases especially the MIT-BIH Arrhythmia database does not provide enough records with an identical combination of ECG-leads, which would be a prerequisite.

**Fourier analysis** At a first glance the analysis of the Fourier spectrum seems like a good candidate for a QRS-feature-extractor. But at a second one, strategies like a simple Fast Fourier Transformation (FFT) [121] or the usage of selected coefficients from a Discrete Fourier resp. Cosine Transformation (DFT resp. DCT) [61] suffer from a major drawback: the loss in time-course information in two ways.

First of all, the calculation of the power spectrum is based on the calculation of the autocorrelation function which, by definition, discards all phase-information.\(^2\) Moreover, Nejadgholi et al. [123] points out that "Signal classification techniques based on power spectral information cannot distinguish between different signals that have the same power spectrum, but such signals may be distinguishable in a reconstructed phase space (RPS)."\(^3\) Even though this thesis will not focus on RPS, the phase information remains an important source of information and discarding it destroys information about the time-course of the signal. Instead, this thesis will concentrate on higher order statistics (HOS).

The second loss in time-course information is caused by the stationarity of the frequency components, which is a consequence of the definition of FFT, DCT and DFT. Luckily there is a method that can analyze changes in time and frequency domain simultaneously on different stages: The Wavelet Transformation (WT). That is why in this thesis no Fourier analysis is carried out and no parameters are used which are derived from the PSD like the spectral entropy [71, 125].

Not using FFT is further supported by findings of Dokur et al. [98]. They used coefficients of Discrete Wavelet and Fourier Transform to analyze the ECG. In their analysis, the DWT coefficients gained 8 percentage points higher accuracy, shorter computation time (80 vs. 130s) and less neural network nodes (11 vs. 18). Even though one decade ago, some authors employed DCT [52, 127] and FFT [62, 86, 128] the Wavelet Transform dominates ECG-feature extraction.

**Hermite functions** Hermite functions are a powerful tool for feature extraction as they provide "a useful parametric model of the QRS-complex [129]" [47]. Each ECG-beat can be considered as a linear superposition of \(N\) hermite polynomials \(H_n\) according to Lagerholm et al. [47]:

\[
x_t = \sum_{n=0}^{N-1} c_n(\sigma) \frac{\exp(-t^2/2\sigma^2)}{\sqrt{\pi} \sigma^n n!} H_n(t/\sigma)
\]

(4.3)

It is advantageous that this orthogonal basis consists not only of mono but also of bi and tri-phasic base-functions. Hence, the determination of the linear coefficients \(c_n\) can easily show basic characteristics of the current arrhythmia or heartbeat. Coefficients close to zero are a hint that a signal is, for example, not bi or triphasic. These coefficients have been employed early for template matching by Sörnmo et al. [129]. More recent approaches by [47] or [41] show promising results using advanced clustering techniques. A recent comparative study by deLannoy et al. [42] revealed that hermite functions and RR-intervals do not supersede common approaches using a combination of RR-interval- and QRS-morphology-features. In their study, a clear drop in the sensitivity of fusion beats manifested.

\(^2\) This problem persists even if methods besides Fourier Transformation are used to calculate the PSD like in [122] \(^3\) Some authors also analyze the phase space via Lyapunov Exponents Güler and Übeyli [32], Übeyli [124], Acharya et al. [125] or the ECG’s trajectory within it [126]

46
might reason that [47] and [41] used more advanced or more fitting classifiers to get the best out of this feature extraction technique. deLannoy et al. [42] uses a support vector machine for classification, just as this thesis. Therefore, the feature is discarded assuming an incompatibility of SVM and features derived from hermite functions.

4.2 Partition functions and relational features

What is feature extraction? To the author, it is simply the description of the striking patterns of a signal utilizing math. In most situations, one has to analyze one dimensional signals, e.g. time series. Even though one can perform several transformation techniques like Wavelet or Fourier transform, the signal stays one-dimensional. Hence, the possibilities for delineation do not increase. In essence, one is left with applying time series analysis to the raw and the transformed signal. This is also true for a special transform: partitioning, i.e. splitting the signal into smaller partitions. Each new partition is again a signal from which features can be extracted.

The advantage of smaller partitions especially in the ECG case is that the raw signal consists of regions of differing information and energy content. The QRS-complex contains lots of information and has a high level of energy. The P-wave, on the other hand, accounts for only a small amount of energy, but still retains information. Moreover, the relation between those partitions can give further insights and can potentially serve as an elaborate feature.

Even though those thoughts will be elaborated on in the subsection about static delineation, the basic feature extraction ideas will be used throughout this chapter. That is why this section intends to provide the framework for the majority of features employed in this thesis which will be exploited in several feature extraction techniques.

This section will first define the two main partitioning strategies and explain the partition functions that can be applied to each of the resulting partitions. It concludes with the definition of relational features, i.e. the difference or ratio between values of the same partition-function resulting from different partitions.

4.2.1 Partitioning strategies

In this thesis two major partitioning strategies are performed: asymmetric and symmetric partitioning. The starting point of each splitting procedure is the vector \( y \in \mathbb{R}^L \) which contains the \( L \) elements of the one-dimensional signal. The basic concepts are shown in fig. 4.1. The symmetric partitioning uses symmetric evenly sized partitions to divide the signal into smaller fractions. The number of partitions \( P \) determines the size of each partition \( \delta \), as \( \delta = P/L \). The asymmetric partitioning method uses accurately defined points to describe the region of interest. In the example below, this is the QT-interval.

From a mathematical perspective, the symmetric way of partitioning \( y \) is performed using the map \( p_s \):

\[
p_s^k(y, P) = \left( y_{\delta(k-1)+1}, y_{\delta(k-1)+2}, \ldots, y_{\delta k} \right) \quad \forall k \in \{1, \ldots, P\}, \quad \delta = \frac{L}{P} \tag{4.4}
\]

The map simply splits each vector \( y \) into \( P \) partitions of size \( \delta = \frac{L}{P} \). This procedure should be applied if
one has only a glimpse about the regions of interest or if the borders of those regions are fuzzy. In the case that one has concrete information about the exact partitions of interest, asymmetric partitioning can be employed. The asymmetric partition map $p_a$ is defined as:

\[ p_a : p_a(y, k_2, k_1) = \left( y_{k_1}, \ldots, y_{k_2} \right) \]  

It simply cuts out the signal elements from index $k_1$ to $k_2$. Thus, the size of the partition is $\delta = k_2 - k_1$.

4.2.2 Partition functions

Now consider a certain partition or even the raw signal. What are the basic properties of those one-dimensional signals? The functions yet to be described are derived from diverse proposals for ECG-classifiers.

It is surely a good idea to measure the minimum and maximum, as well as the difference between minimum and maximum of the partition of size $\delta$, now denoted as $y$:

\[ \min y : \min_i y_i \]  
\[ \max y : \max_i y_i \]  
\[ \text{maxmin} y : \max y - \min y \]
One would also like to get information on where the signal reaches its extrema and how far they are apart from each other. Hence, the computation of the arguments of the minimum and maximum and their respective distance is necessary.

\[
\arg \min y: \ arg \min_i y_i \tag{4.9}
\]

\[
\arg \max y: \ arg \max_i y_i \tag{4.10}
\]

\[
\arg \text{minmax} y: \ arg \max y - \ arg \min y \tag{4.11}
\]

All these six measures are also performed for \(\text{abs}(y)\), i.e. the vector of the absolute values of the elements in \(y\). This should allow the investigation of the absolute amplitudes of signals which are not restricted to be either strictly positive or negative.

Another very basic assessment is the calculation of the mean, median, standard deviation and variance:

\[
\mu(y): \ \frac{1}{\delta} \sum_{i=1}^{\delta} y_i \tag{4.12}
\]

\[
\mu_{\text{med}}(y): \ \frac{y_{\text{sort}}^{\delta+1}}{2} \tag{4.13}
\]

\[
\sigma(y): \ \sqrt{\frac{1}{\delta - 1} \sum_{i=1}^{\delta} (y_i - \mu(y_i))^2} \tag{4.14}
\]

\[
\sigma^2(y): \ \frac{1}{\delta} \sum_{i=1}^{\delta} (y_i - \mu(y_i))^2 \tag{4.15}
\]

The variance is in part a measure of the energy content of the signal. This is typically assessed using the average power (\(\text{AP}\)) or the mean of the absolute values (\(\text{MA}\)):

\[
\text{AP}(y) = \mu(\ y^2) \tag{4.17}
\]

\[
\text{MA}(y) = \mu(\ \text{abs}(y)) \tag{4.18}
\]

All the aforementioned partition functions assessed pretty basic aspects of the signal. Now let us investigate on the shape of the distribution of the values in the partition a little bit more. First off, one can perform a linear regression which yields the slope and the intercept of a line fitting \(y\).
Two fairly informative measures are the variance of the auto-correlation and the Shannon entropy, which basically assesses if the signal shows a uniform distribution.

- standard deviation [32, 130–132], variance, average power [32, 131]
- median, mean of absolute values [32, 131], mean [132]
- variance of auto-correlation [114], shannon-entropy [67] (see [133, 134] for more details)
- minimum [132], maximum [132, 135], minmax, position of minimum and maximum, maximum and minimum of absolute values
- slope and intercept of linear regression

4.2.3 Relational features

In many circumstances calculating partition functions on certain partitions alone is not sufficient. One has to take the ratio or difference between the outcome of certain partition functions on different partitions into consideration. For instance, the duration of the QT-interval should be measured with respect
4.2 Partition functions and relational features

to the QRS-width. There are infinite possibilities where the ratio between two parameters tells more than the parameters alone. Therefore, this thesis will calculate relational features for the majority of partitioning based features.

Suppose one wants to analyze a certain signal $y$, which can surely be a transformed or even previously partitioned raw signal. Then one applies a single or a set of partition maps which yield $N$ different partitions $\pi_i$ derived from the signal $y$. For any partition function $f(\pi_i)$ one can define a relational feature matrix $\Lambda$ using the relation $\rho(a,b)$:

$$
\Lambda_i(\rho, f) = \rho \left( f(\pi_i), f(\pi_i) \right)
$$

This thesis will use $\rho_{\text{Div}}(a,b) = a/b$ and $\rho_{\text{Sub}}(a,b) = a - b$ only. Moreover, with regard to the latter relation, only the elements of the upper triangular matrix $\Lambda_{ij}$, $i < j$ are used as features. Hence, the number of relational features using subtraction is $\frac{1}{2}N(N-1)$. Most of the feature selection techniques introduced later on rely primarily on sorting the values of a certain feature. Therefore, a new feature should possess an order different from its predecessors. Clearly, features resulting from $a - b$ will have the reverse order than of features derived from $b - a$. In consequence, their order is practically the same. As

$$
\Lambda_i(\rho_{\text{Sub}}(a,b), f) = \Lambda_i(\rho_{\text{Sub}}(b,a), f)
$$

holds, the lower triangular matrix can be omitted. With respect to the division relation $\rho_{\text{Div}}$, the situation changes. Consider a simple example where $a = (1, 0.5, 1)$ and $b = (-1, -1.1)$. $\rho_{\text{Div}}(a,b)$ yields $(-1, -0.5, 1)$, whereas $\rho_{\text{Div}}(b,a)$ yields $(1, -2, 1)$. Hence, the order is not reversed and the first two numbers only change places, while the last number maintains its position. In fact, only with respect to their absolute values, sorting $\rho_{\text{Div}}(|b|, |a|)$ will have the reversed order of $\rho_{\text{Div}}(|a|, |b|)$. As strict positivity of all $a$ and $b$ can not be ensured, the whole matrix $\Lambda$, despite the diagonal $i = j$, is analyzed. Thus, $N(N-1)$ features are extracted using relational features derived by $\rho_{\text{Div}}$.

The calculation of the raw values alone will not necessarily yield satisfactory features. This is due to the ECG’s high inter and intra-patient variability. One assumption of this thesis is that even though the absolute values do change, the relative values stay stable. This can be easily seen by taking a closer look at the ECG-amplitudes. Even though the P-waves and QRS-complexes will have changing raw amplitudes, the ratio between the amplitude of the P and R-wave should remain untouched. Hence, in many situations the ratio or difference between related parameters is calculated. Thus, for two parameters $a$ and $b$ one also computed $a-b$, $a/b$ and $b/a$. The reason for omitting $b-a$ is simple. The majority of feature selection techniques to come will rely on sorting the values for each feature. As $b-a$ will have simply the reversed sorting of $a-b$, there is no point in utilizing its inverse. Given a group of $N$ features the amount of relational features generated from this group is:

$$
N_{\text{rel}} = \frac{N(N-1)}{2} + (N^2 - N) = \frac{3}{2} \cdot N(N-1)
$$

This can be easily verified with regard to a $N \times N$ matrix, which holds the ratio or difference between the $i^{th}$ and $j^{th}$ feature in the group. Obviously the number of relational features according to $a - b$ is given by the number of elements in the upper triangular matrix: $N(N-1)/2$. As the ratio between features in the group is calculated using $a/b$ and $b/a$, both, the upper and lower triangular matrix, are utilized. In
consequence, the whole matrix is used without its diagonal, yielding $N^2 - N$.

### 4.3 QRS-morphology

One of the most important features in the classification of ECG-patterns are features describing the QRS-morphology. In a recent study, Zadeh et al. [68] showed that removing morphologic features and using only features based on RR-intervals, results in a drop of accuracy by over 50 percentage points. Whereas if they keep the morphology and discard the RRI features, accuracy decreases by 3.5 percent points only. One should not generalize those findings by focusing only on QRS-morphology features, but they surely are the backbone of any feature-set that will be successful in ECG pattern classification.

The major strategy for describing the morphology of the QRS-complex is to find a mathematical algorithm that extracts the characteristic points (PQRST) of the heartbeat and compute several statistical measures using these points. This process is called ECG-delineation and is widely applied in the literature on ECG-morphology analysis. This section will give a summary of the current attempts for ECG-delineation and the parameters that are derived from the characteristic points of the ECG. Even though these parameters show satisfactory results by means of classification accuracy, there seems to be a big loophole for misclassification. The reason for the misclassification lies in the epistemological concept of those algorithms. The basic idea is that by emulating the cognitive process of the cardiologist (find PQRST, get some variables out and put it into a neural network), one will have a good chance of getting similar classification results. The problem with this approach is that it is most likely to be valid only for physiologic heartbeats. In pathologically shaped ECG-segments, the definition of the PQRST points will be misleading, e.g. the fusion of ventricular and normal beats will cause the P-wave to be superposed by the fused ventricular excitation. From the author’s point of view it is therefore necessary to employ a "static ECG-delineation" complementary to the "physiologic ECG-delineation" mentioned above. This second approach, developed in this thesis, will be explained in detail in the second part of the section.⁴

#### 4.3.1 Physiologic delineation

As described in the QRS-delineation section, the accuracy of the delineation is a big problem. This is reflected within the recommendations for QRS-delineation of The CSE working party [136]. Even cardiologists do not agree on the precise point of P-wave on and offset. Standard deviation is 10.2 ms for the onset and 12.7 ms for the offset. This problem is even more pronounced for T-waves. The end of the T-waves has standard deviation among cardiologists’ consensus of around 32.5 ms. One can conclude that those points are not strictly defined and easy measurable features. Maybe this is one reason why feature selection methods like SFFS (see [40]) drop physiologic delineation features and favor wavelet parameters and logarithmic RR-interval measures.

Moreover, the question persists, if describing the QRS-complex by physiologic features yields a set of independent features. Findings of Christov and Bortolan [16] raise doubts. They showed that in the case of VPC vs. non-VPC classification amplitude, area, interval, slope and vector-cardiography features yield comparable sensitivities, whereas specificity for amplitude measures is best.

Now, let us suppose the PQRST-points have been determined with a satisfying amount of accuracy by the ECGpuwave algorithm. In the paragraphs to come, several approaches to physiologic delineation are...
discussed. The term physiologic delineation refers to any feature extraction strategy intending to derive features based on points with physiologic meaning. The last paragraph will summarize the majority of them into a single mathematical framework.

**QRS-timing features** The duration of the R-S interval \( [116] \), R-T interval \( [116] \), standard Q-T interval \( [116] \) or corrected by Frieryda et al. \( [24, 137] \) are used for basic timing measures. The prolongation of the waves themselves are also a common feature: the width of the whole QRS-complex \( [5, 7, 16, 24, 35, 58, 63–66, 71, 83, 99, 111, 117, 137–140] \), the duration of the T-wave \( [7, 24] \) and of the P-wave \( [99, 101] \)\(^5\). de Chazal and Reilly \( [83] \) use the duration between the onset and offset of the Q,R,S segments. In the majority of publications, the durations are measured between the onset and offset of the QRS-complex respective to other intervals like Q-T or S-T \( [99] \). An exception can be found in \( [83, 101] \) where the P-R interval is measured between P-wave onset and R-peak onset and the Q-T interval between offset and offset.\(^6\) With regard to the QRS-width, not all authors use the same calculation method\(^7\) and some like Osowski and Linh \( [117] \) use parameters only corresponding to the QRS-width. For more information see \( [88, 89] \) and section QRS-complex detection and delineation.

**QRS-amplitude features** A widely used feature is the amplitude of the R-peak \( [24, 35, 68, 83, 99, 107, 113, 116, 137] \), which is often accompanied by the measurement of the peak amplitudes for the P, S, Q and T waves \( [24, 68, 83, 99, 113, 137, 141] \). Chudácek et al. \( [24, 137] \) moreover use the ratio between the different amplitudes. Besides the amplitudes, Zadeh et al. \( [68] \) propose to use the positions of the PQRSST points of the ECG-segment. A combination of amplitudes and position can be found in \( [141] \), which defines the "Q-wave mass center".

**Area features** Furthermore, the area under the QRS-complex is often analyzed \( [7, 83, 107, 111, 116] \). It serves as a surrogate for the amount of activated heart muscle during the course of excitation and is sometimes calculated over the whole beat segment \( [7] \). Moreover, some authors use the area between Q-R, R-S \( [83] \), S-T \( [116] \) and Q-T \( [101] \). Most of the areas are calculated simply on the amplitudes of the ECG. This can be somehow misleading in case of baseline wander, therefore de Chazal and Reilly \( [83] \) calculate the area of the P-wave relative to the baseline. Another approach by Moraes et al. \( [138] \) calculates the area as sum of absolute distances of each sample point from the first QRS-sample-point and as sum of adjacent absolute differences in the QRS-complex.

**Slope features** The latter parameters focus mainly on the amplitudes, but do not characterize the time-course of the QRS-complex. Therefore, few authors provide additional features that analyze the linear regression from the onset to the R-peak \( [7, 16, 65, 66] \) or the slope of the S-T-segment \( [116] \). A quite rare feature is regression slope between two adjacent peaks \( [65, 66] \). Even though slope features are particularly employed for morphology analysis, the normalized slope in a sliding 10 samples window could be a potent feature for QRS-detection\( [33] \).

**Spectral analysis** The features summarized so far are time-domain measures. Even though there should be complementary information in the frequency-domain, features making use of fourier-analysis are rare. Hosseini et al. \( [116] \) use the mean spectral power density and the auto-correlation coefficient that is used by Ghongade and Ghatol \( [107] \), too\(^8\). With regard to Parseval's theorem that the power...
of the Fourier-transform is equal to the power of the signal itself, one can regard the QRS-power and beat power measures proposed by Mar et al. [7] as features of spectral analysis. The QRS-power is also used by Moraes et al. [138], where it is denoted as total amplitude of the QRS. The reason for the low attention that is payed to fourier analysis is supposedly the fact that Wavelet transformation is considered to be much more powerful for analyzing the time-frequency course of the ECG.

A unified approach The ECGpuwave algorithm yields the positions of the centers of the PQRST-points \( k_P, k_Q, k_R, k_S \) and \( k_T \). Moreover, it provides the positions of the onset and offset of the T-wave \( k_{onT}, k_{offT} \). Therefore, one major feature is the amplitude at each of those points, denoted as \( amp(i') \).

Let us recall that \( y'_i \) denotes the whole segment of the \( i^{th} \) heartbeat of length \( \Delta_{seg} \). This thesis defines 15 different physiologic partitions using the map \( p(k_2, k_1, y) \), which extracts the components between the indices \( k_1 \) and \( k_2 \) (see eq. 4.5). Four partitions are mainly concerned with the P-wave: \( k_{onP} - k_P - k_{offP} \). \( k_{onP} - k_{offP} \), \( 1 - k_P \). Partitions analyzing the QRS-complex are: \( k_Q - k_R - k_S \), \( k_P - k_R \), \( k_Q - k_S \). The latter are accompanied by seven more, which account for the T-wave: \( k_Q - k_T \), \( k_S - k_T \), \( k_S - k_{onT} \), \( k_{onT} - k_T - k_{offT} \), \( k_{onT} - k_{offT} \), \( k_T - \Delta_{seg} \).

All 22 partition functions are applied to each of the 15 partitions. Moreover, the duration of each segment is measured using the number of elements \( \delta \) of the partition vector. A conclusive summary about the number of features used for physiologic delineation can be found in section 4.8.

### 4.3.2 Static delineation

A first attempt of unifying physiologic delineation approaches was carried out by de Chazal and Reilly [83]. They used functions on defined segments, where area, amplitude and duration of each Wave Q, R, S were measured. Yet they did not cover a larger amount of features and used only features that a cardiologist would come up with. As described in the latter subsection, choosing \( k_P, k_Q \) etc. by physiologic delineation causes severe problems. This is due these points being sometimes ill-defined.\(^9\)

In a later publication, de Chazal et al. [38] could show that given single lead parameters, so called "fixed-interval ECG-morphology features", show higher specificity and accuracy, whereas "segmented morphology features" show increased sensitivity. This corroborates the statement that a static delineation accompanied by physiological delineation should yield a powerful feature set. In their approach, they only used the amplitudes in either fixed intervals or physiologically delineated intervals. A more recent study by deLannoy et al. [42] used a quite similar approach by defining a small amount of partition functions (6) according to Christov et al. [65] and applying them to physiologically defined segments. The approach developed in this thesis will use the unified strategy for physiologic delineation also proposed herein and transfer it to fixed intervals.

**Symmetric ECG-partitioning** The ECG-partitioning is slightly different in static delineation, than it was for physiologic. Again the main goal is to split the segment containing the \( i^{th} \) preprocessed heartbeat \( y'_i \). This is done by slicing the segment into \( P \) equally sized partitions using eq. 4.4.\(^{10}\)

\(^9\) Therefore, de Chazal and Reilly [83] introduce a parameter that measures absence of P-wave (+ or -).\(^{10}\) [38] simply chooses 10 out of a 250 ms segment asymmetric around the fiducial point and 8 samples in a segment of 350ms at the position of the T-wave
4.3 QRS-morphology

The number of partitions determines the granularity of the analysis and whether the partitions analyze the edges or the center of the QRS-complex. If \( P \) is odd the partition \( P/2 + 1 \) contains the QRS-complex, because \( y'_i \) is centered around the fiducial point. If \( P \) is even it is more likely to analyze the edges of the QRS-complex. Moreover, the analysis depends on the segment size \( \Delta_{seg} \): Given a fixed \( P = 5 \) the width of the QRS-snippet to be analyzed is given by the amount of samples comprising the vector \( p_i(\Delta_{seg}, 5) \). This facilitates the selection of ECG-snippets of interest. As already mentioned in the ECG-segmentation stage, there are 3 segments of interest: the narrow QRS-segment (±75 samples), the wider T and P-wave containing segment (±150 samples) and the segment in between the QRS-samples (−250 samples). The partitioning now allows for the selection of specific regions in each of those segments and the comparison of those regions to each other and different segmentation strategies using defined partition functions.

The partition functions \( \tilde{\pi} \) are the core of the static delineation because they calculate the statistics in each partition and compare them to each other. With the exception of duration, all partition functions used during physiologic delineation will also be used within static delineation. Other authors, moreover, use the sum of absolute values as well as the sum of all negative resp. positive amplitudes.\[^{65, 66}\] These functions are not considered highly sensitive or specific. Hence, they are discarded.

In this thesis, 32 different partitions are used (\( P \in \{3, 5, 8, 16\} \)). After calculating the \( j^{th} \) partition function of the \( k^{th} \) partition of the \( i^{th} \) beat according to:

\[
\mu_{ikj}; \begin{cases} 
\tilde{\mu}_j(p_i(\Delta_{seg}, 3)) & 0 < k \leq 3 \\
\tilde{\mu}_j(p_{i-3}(\Delta_{seg}, 5)) & 3 < k \leq 8 \\
\tilde{\mu}_j(p_{i-8}(\Delta_{seg}, 8)) & 8 < k \leq 16 \\
\tilde{\mu}_j(p_{i-16}(\Delta_{seg}, 16)) & 16 < k \leq 32 
\end{cases} \quad (4.25)
\]

\[
\begin{align*}
\tilde{\mu}_{ij} = \frac{\mu_{ikj}}{\mu_{ikj}'} \\
D_{ij} = \mu_{ikj} - \mu_{ikj}'
\end{align*} \quad (4.26)
\]

The obtained values will be used to calculate further relational features, i.e. the difference \( D_{ik}^{ij} \) and the ratio \( R_{ik}^{ij} \) in between the partitions\[^{\text{11}}\].

\[
R_{ik}^{ij} = \frac{\mu_{ikj}}{\mu_{ikj}'} \quad (4.27)
\]

\[
D_{ik}^{ij} = \mu_{ikj} - \mu_{ikj}' \quad (4.28)
\]

4.3.3 QRS-asymmetry

The features described in the latter subsection are intended to be pretty robust with regard to noise and inter- and intra-patient variability due to their pure statistical characteristic. A major problem is that those parameters could probably be too unspecific. Besides standard QRS evaluation techniques described before, this thesis introduces a new approach for measuring the QRS asymmetry that has not been discussed in literature before. The basic idea is quite simple: To measure the asymmetry of the QRS-complex one has to measure the crossing points of the QRS-complex with a horizontal line at a specific level. The distance from the right crossing point to the R-peak is the right asymmetry and the

\[^{\text{11}}\] Mar et al.\[^{7}\] use the ratio between the beats' minimum and maximum amplitude. Still their approach remains unsystematic.
distance from the left crossing point to the R-peak is the left asymmetry. The major problem is that the
QRS-complex is not as well as defined as one would wish. Hence, the definition of the crossing points
needs clarification. Moreover, the height of the QRS-complex varies from beat to beat, which in this case,
is not much of a problem because in the preprocessing-stage the signals are already normalized between
zero and one.

With the normalized signal at hand, the definition of the crossing point is no problem: The measurement
of the asymmetry of a certain ECG-beat \( y_i' \) starts by selecting the level \( \alpha = 0.9 \) at which the asymmetry
should be measured. With this level, one can define the set of zero-crossings \( \tilde{Z}_i(\alpha) \):

\[
\tilde{Z}_i(\alpha) = \left\{ y_k \in \tilde{Z}_i(\alpha) \mid \frac{y_{k+1} - \alpha y_k}{y_k - \alpha} < 0, k \in \{1, \ldots, \Delta_{seg} - 1\} \right\}
\] (4.29)

Despite the fact that the number of zero-crossings \( Z_i(\alpha) \) will be used as a feature, it is exploited for the
definition of the right and left zero-crossing (\( Z_r(\alpha) \) resp. \( Z_l(\alpha) \)):

\[
Z_i(\alpha) = |\tilde{Z}_i(\alpha)|
\] (4.30)

\[
Z'_r(\alpha) = \min_{z \in Z_i(\alpha)} z - \frac{\Delta_{seg}}{2}
\] (4.31)

\[
Z'_l(\alpha) = \min_{z \in Z_i(\alpha)} z + \frac{\Delta_{seg}}{2}
\] (4.32)

\[
Z''(\alpha) = Z'_r(\alpha) - Z'_l(\alpha)
\] (4.33)

The following measures are used conclusively: The number of zero-crossings \( Z_i(\alpha) \), the left asymmetry
\( \left( \frac{\Delta_{seg}}{2} - Z'_l(\alpha) \right) \), the right asymmetry \( \left( Z'_r(\alpha) - \frac{\Delta_{seg}}{2} \right) \) and the total width of the QRS-complex \( Z''(\alpha) \) measured
at \( \alpha \in \{0.1, 0.25, 0.5, 0.75, 0.9\} \) levels. As previously described for the statistical measures, the ratio and
differences of the features at different \( \alpha \) levels are used for the final feature evaluation as well.

4.4 Wavelet transform

Even though the analysis of the QRS morphology in the time-domain is very important for ECG-classification,
one surely needs to extract information about the frequency domain of the signal to gather as much in-
formation as possible about the morphology at hand. A typical approach would be to analyze the fourier
spectrum of the signal. Unluckily, the frequency information retained by a Fourier transform (FT) is
not localized in time, i.e. one does not gain any information of the variation of amplitudes of certain
frequencies over time. Therefore, the short-time fourier transform (STFT) introduces a windowing func-
tion that performs the time-localization. But still, the uncertainty principle prevents one from having
perfect time and frequency resolution simultaneously, i.e. the product of frequency and time-resolution
is fixed.

For the analysis of the ECG-signal, one would like to have a good time resolution for fast changes (high-
frequency) and a good frequency resolution for the slow variations or trends (low-frequency) of the
signal. As long as the good time resolution is traded for a worse frequency resolution, one does not
violate the uncertainty principle. Such a setup can not be implemented using STFT because time and frequency resolution is fixed for all frequencies.

The solution is depicted in fig. 4.3. The wavelet transformation (right side) has high time resolution for fast changes, but is quite uncertain about their frequency. On the other hand, it can accurately determine the frequency of slow changes, but at the expense of low time resolution.\footnote{This is a common situation in many biomedical applications. Hence, the wavelet transform is widely employed. See [142] for a brief overview.}

In the first part of this chapter it will be discussed, why DWT is superior to STFT for this application. The second part will give a brief summary about the analogy between DWT and multi-scale analysis (MSA). From a practitioner’s view point, DWT is a composition of FIR filters. Therefore, in the remaining part of this chapter, DWT is discussed only with regard to the underlying FIR filters and their properties. In the third part, the four major families of orthogonal wavelets are presented, together with their filter properties.

4.4.1 Continuous wavelet vs. fourier transform

For a better understanding of what the continuous wavelet transform (CWT) is and how it relates to STFT, let us first start with some helpful definitions\footnote{The following derivations to the basic differences between short-time Fourier transform and wavelet transform are according to [143]}:

Let $a(n)$ be an element of the space of square-summable sequences $l^2(\mathbb{Z})$ and $f(x)$ an element of the space of square-integrable functions $L^2(\mathbb{R})$. Hence, both spaces are Hilbert spaces and one can formulate the metric of those spaces using the inner product:
Chapter 4 Feature Extraction

\[
\langle a(n), b(n) \rangle = \sum_{-\infty}^{\infty} a^*(n)b(n) \quad ||a(n)||^2 = \langle a(n), a(n) \rangle
\]

\[
\langle f(x), g(x) \rangle = \int_{-\infty}^{\infty} f^*(x)g(x)dx \quad ||f(x)||^2 = \langle f(x), f(x) \rangle
\]

(4.34)

With those definitions at hand, one can easily define the fourier transform of a continuous time signal \(y(t)\):

\[
\hat{y}(\omega) = \langle e^{i\omega t}, y(t) \rangle = \int_{-\infty}^{\infty} e^{-i\omega t}y(t)dt
\]

(4.35)

The major problem when using the fourier transform is that "because of the infinite extent of the basis functions, any time-local information (e.g. abrupt change in the signal) is spread out over the whole frequency axis." [143] To address this problem, the STFT introduces a windowed Fourier transform, employing the window function \(w(t - \tau)\), which leads to a more time-localized version of the fourier transform:

\[
\hat{y}(\omega, \tau) = \langle e^{-i\omega t}w(t - \tau), y(t) \rangle
\]

(4.36)

As \(\omega\) is still independent from \(\tau\), frequency and time resolution remain fixed. In the framework of the discrete wavelet transform (DWT), both resolutions are linked. The DWT is derived from the continuous wavelet transform:

\[
y(a, b)_{WT} = \langle \psi_{a,b}(t) | y(t) \rangle \quad \text{with} \quad \psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left( \frac{t-b}{a} \right) \quad |a| \in \mathbb{R}^+, b \in \mathbb{R}
\]

(4.37)

by discretization, i.e. setting \(a = a_0^m\) and \(b = nb_0a_0^m\):

\[
\psi_{mn}(t) = a_0^{-m/2} \cdot \psi(a_0^{-m}t - nb_0) \quad m, n \in \mathbb{Z}, \quad a_0 > 1 \quad b_0 \neq 0
\]

(4.38)

Referring to eq. 4.37 and fig. 4.3 the dilation \(a\) determines the frequency of the wavelet, so wavelets with high dilation have more oscillations and are shorter. \(b\), on the other hand, describes the time-shift of the wavelet. The discretization of the DWT links the time shift and the oscillations, which ensures that "the translation step depends on the dilation, since long wavelets are advanced by large steps, and short ones by small steps"[143].

**Constraints of the mother wavelet**  The wavelet according to the definitions above are dilated and scaled versions of the so called "mother wavelet". Obviously, not all functions are potential "mother wavelets". They have to meet certain criteria. Two very important ones are [144, 145]: (1) \(\psi(t)\) having finite energy and (2) zero mean. The first constraint tells us that \(\psi(t)\) has to be an element of \(L^2(\mathbb{R})\) and therefore \(\langle \psi(t) | \psi(t) \rangle < \infty\). Constraint two directly follows from the admissibility condition:

\[\text{typically } w \text{ is a Gaussian}\]

58
4.4 Wavelet transform

\[ C_g = \int_0^\infty \frac{\hat{\psi}(\omega)}{\omega} d\omega < \infty \quad (4.39) \]

which restricts the Fourier transform of the wavelet \( \hat{\psi}(\omega) \) to be regular and that \( \hat{\psi}(w = 0) = 0 \) from which \( \hat{\psi}(t) = 0 \) directly follows. The admissibility condition ensures energy conservation by means of the Parseval theorem:

\[ E = \int_{-\infty}^{\infty} d\omega |\hat{y}(\omega)|^2 = \int_{-\infty}^{\infty} dt |y(t)|^2 \quad (4.40) \]

It simply states, that the \( L^2 \)-norm is invariant under Fourier transformation. Now let us reformulate this in bra-ket notation, in which \(|\hat{y}(\omega)\rangle = (e^{i\omega t} |y(t)\rangle | e^{i\omega t}\rangle\).

\[
E = \int_{-\infty}^{\infty} d\omega |\hat{y}(\omega)|^2 = \int_{-\infty}^{\infty} d\omega (e^{i\omega t} |y(t)\rangle | e^{i\omega t}\rangle)(e^{i\omega t} |y(t)\rangle | e^{i\omega t}\rangle) \\
= \int_{-\infty}^{\infty} d\omega (e^{i\omega t} |y(t)\rangle | e^{i\omega t}\rangle)(e^{i\omega t} |y(t)\rangle) \\
= \int_{-\infty}^{\infty} d\omega \langle y(t), e^{i\omega t\rangle} \langle e^{i\omega t}, y(t)\rangle = \langle y(t) | y(t)\rangle = \int_{-\infty}^{\infty} dt |y(t)|^2 \quad (4.43) 
\]

This derivation\(^{15}\) shows that the \( L^2 \)-norm of \( y(t) \) is invariant under change of basis, e.g. the continuous Fourier transform.

With regard to Proposition 5.1 in [146] "any \( f(t) \) from \( L^2 \) can be written as a superposition of shifted and dilated wavelets." The wavelet expansion of a function \( f(t) \) looks like:

\[ f(t) = \frac{1}{C_g} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \langle \psi(a, b) | f \rangle \psi(a, b)(t) \frac{da \, db}{a^2} \quad (4.44) \]

This equality only holds because \(| \psi(a, b) \rangle \) is a basis of \( L^2 \). More precisely it forms a tight frame with \( C_g \), measuring the redundancy of the basis.[147].

\(^{15}\) To be more elaborate, the basis \(| e^{i\omega t}\rangle\) is only defined on a pre-Hilbert space using the inner product \(|f(t) | g(t)\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f^*(t) \cdot g(t) dt\).
4.4.2 Orthogonal wavelets form a basis of $\mathcal{L}^2$

The major drawback of the continuous wavelet transform is its high redundancy $C_g > 1$, its expensive computation (integral $\langle \psi(a,b) | y(t) \rangle$) and the arbitrary choice of $a$ and $b$. The discrete wavelet analysis gets rid of those issues. As most wavelets are defined on a dyadic grid, one should stick to setting $a_0 = 2$ and $b_0 = 1$, which yields the scaling function of the discrete wavelet transform (DWT):

$$\phi_{mn}(t) = 2^{-m/2} \cdot h(2^{-m}t - n) \quad m, n \in \mathbb{Z} \quad (4.45)$$

This scaling function is the father wavelet and should not be confused with the mother wavelet $\psi_{mn}$, which is described later on. Unlike the mother wavelet, the scaling function is constrained to have mean one

$$\int_{-\infty}^{\infty} \phi(t) dt = 1 \quad (4.46)$$

and be self-similar in the sense of:

$$\phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi(2t - k) \quad (4.47)$$
$$\phi_{00}(t) = \phi(t) \quad (4.48)$$

This means that a scaling function has to be a superposition of dilated and shifted versions of its own. The scaling function $\phi_{mn}$ is not a basis of $\mathcal{L}^2$ because clearly $\langle \phi_{mn} | \phi_{m'n'} \rangle \neq \delta_{mm'} \delta_{nn'}$. Nonetheless, it forms a basis of the subspace $V_m \subset \mathcal{L}^2$, hence $\langle \phi_{mn} | \phi_{m'n'} \rangle = \delta_{mm'}$. There exists a complementary subspace $W_m \cap V_m = \emptyset$, which has an orthonormal basis derived from the scaling function.

$$\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \phi(2t - k) = \sqrt{2} \sum_{k \in \mathbb{Z}} (-1)^k h_{1-k} \phi(t - k) \quad (4.49)$$
$$\psi_{mn} = \sum_{k \in \mathbb{Z}} g_k \phi_{m-1,k} = \sum_{k \in \mathbb{Z}} (-1)^k h_{1-k} \phi_{m-1,k} \quad (4.50)$$

A nice property of the $W_m$ subspace is that $V_{m+1} = V_m \oplus W_m$. Hence, $\mathcal{L}^2$ is divided into a cascade of embedded subspaces, i.e.

$$\mathcal{L}^2 = V_0 \oplus \sum_{m=0}^{\infty} W_m \quad (4.51)$$

As $\langle \psi_{mn} | \psi_{m'n'} \rangle = \delta_{mm'} \delta_{nn'}$, the complementary subspaces $W_m$ are orthogonal to each other. Therefore, the basis of $\mathcal{L}^2$ is $\psi_{mn}$. In conclusion, any scaling function satisfying the above constraints can be
extended to form a basis (the mother wavelets) of $\mathcal{L}^2$.

**An example**  Let $A_m y(t)$ be an approximation of a certain signal $y(t)$, i.e. sampling the data with the scaling functions $\phi_{mn}$ at level $m$. The digital sampling of a signal is equal to low-pass filtering because changes above the sampling frequency are dampened. Hence, $A_m y(t)$ is the best approximation of the signal. Its representation in the wavelet basis is:

$$A_m y = \sum_{n \in \mathbb{Z}} \langle \phi_{mn} | y \rangle \phi_{mn} \quad (4.52)$$

One can decompose this approximation using the orthonormal basis $\psi_{mn}$ by blurred and differentiated versions (see illustration in fig. 4.4) of the previous approximation according to:

$$A_{m_0} y = A_{m_0} y + \sum_{m=m_0+1}^{m_1} \sum_{n \in \mathbb{Z}} \nu_{mn} \psi_{mn} \\ D_m y \quad (4.53)$$

Figure 4.4 illustrates that with each approximation the blurred signal has lower time resolution and the difference version shows higher time resolution, but contains a broad frequency spectrum.
4.4.3 Orthogonal wavelets act as perfect reconstruction filter banks

The last section showed that DWT has better time and frequency resolution in the frequency bands resp. timespans of interest. This is due to their basis functions linking time and frequency resolution. Low frequency wavelets have a high time step, whereas high frequency wavelets have a low time step. Those wavelets are not arbitrarily chosen. A signal can be decomposed into its wavelet coefficients and perfectly reconstructed afterwards. Therefore, the wavelet functions are a real basis in $L^2$.

In the design of digital filters, perfect reconstruction is well established\(^\text{16}\). One only needs two low-pass filters $A, \tilde{A}$ and two high-pass filters $D, \tilde{D}$. As long as they satisfy:

$$A(-z)\tilde{A}(z) + D(-z)\tilde{D}(z) = 0$$  \hspace{1cm} (4.54)
$$A(z)\tilde{A}(z) + D(z)\tilde{D}(z) = 2$$  \hspace{1cm} (4.55)

Perfect reconstruction is guaranteed in all circumstances. A schematic of a perfect reconstruction bank is depicted below.

Now the equation system for the PR-FB can only be solved by setting:

$$D(-z) = z\tilde{A}(A(z))$$  \hspace{1cm} (4.56)
$$\tilde{D}(z) = -z^{-1}A(-z)$$  \hspace{1cm} (4.57)

This leaves only the choice of $l$ being odd. The above equation system can be summarized: One easily checks that this is equal to finding a filter $M(z) = A(z)\tilde{A}(z)$ satisfying:

$$M(z) + M(-z) = 2$$  \hspace{1cm} (4.58)

In the case of orthogonal wavelets ($D(z) = \tilde{D}(z^{-1})$) the synthesis filter banks $\tilde{A}, \tilde{D}$ are just the time

\(^{16}\) For a detailed analysis about compactly supported orthogonal wavelets, see the original work done by [147–150]
reversed version of the analysis filter banks $A$, $D$. Applying the time reversal yields:

$$D(z^{-1}) = -z^{-l}A(-z)$$

(4.59)

If $D(z) = g_kz^{-k}$ and $A(z) = h_kz^{-k}$ one retains:

$$g_k = (-1)^k h_{l-k}$$

(4.60)

It is apparent from the above derivations that there are two ways of describing orthogonal wavelets. The first one is finding a scaling function satisfying the similarity and mean one criteria. The “similarity” coefficients $h_k$ are then exploited to build the mother wavelet using the above formula. Any signal can now be represented by approximations and differences of the signal using the mother wavelets as the basis of $L^2$. The second approach is finding a combined filter $M(z)$ satisfying eq. 4.58 and use the coefficients $h_k$ again to create the corresponding high pass filters $D$ and $\tilde{D}$. As the latter viewpoint is much easier to handle, one should stick to it.

4.4.4 Discrete wavelet decomposition: Mallat’s scheme

So how does one implement the multi-scale analysis using the filter banks defined above? This is straightforward. As the analysis filters $A_0$ and $D_0$ form a PR-FB, they can also be exploited to analyze their own output. This cascade of filters is called Mallat’s Multi-scale Analysis Scheme (MSA) [151, 152] and is depicted in fig. 4.6.

Given the original signal $y(t)$, the low-pass filter $H$ removes frequency content $f > f_{Ny}/2$ "generating the first order average" $A_1$ by subsequently down-sampling (removing even indexes) the filtered data.\footnote{$f_{Ny}$ denotes the Nyquist frequency, which is half the sampling frequency.} Ideally the low-pass removes all frequency information $f > f_{Ny}/2$ and the high-pass all frequency content $f < f_{Ny}/2$. A perfect filter with a precise cut-off at $f_{Ny}$ would have infinite filter length, which decreases time-resolution.
Chapter 4 Feature Extraction

The same procedure is performed using the filter $G$ and a down-sampler, yielding the “first order difference” $D_1$, which retains the frequency content $f > f_{Ny}/2$.

This scheme ensures high time resolution for high frequencies because $D_1$ comprises half of the signal duration, even though the frequency resolution drops, because the half spectrum is stored in $D_1$. Moreover, high frequency resolution for low frequencies is assured because $A_1, A_2$ and $A_3$ comprise more and more narrow bands of the spectrum, while the temporal resolution drops because the averaged signal contains less and less of the total signal duration.

Let us take a look at a simple signal decomposed according to Mallat’s scheme using the Haar wavelet. From the perfect reconstruction properties of the DWT, one can deduce that knowing $A_2 y$, $D_2 y$ and $D_1 y$ is sufficient for reconstructing $A_0 y$. With regard to the Haar wavelet, the low-pass filter is a simple moving average $H = [0.5 0.5]$ and the high-pass filter is simply the difference $G = [−0.5 0.5]$. Hence, the reconstruction is facilitated. The first approximation $A_1 y = A_2 y + D_2 y$ and the original signal $A_0 y = A_1 y + D_1 y$.

4.4.5 Stationary Wavelet Transform (The à trous scheme)

So far, the discussion focused on orthogonal wavelets. Orthogonality is a nice property. However, the superiority of orthogonal wavelets for the purpose of feature extraction, compared to non-orthogonal ones, has not been proven yet. Orthogonal ones do provide a very dense representation of the time-frequency behavior of the ECG. This comes at the expense of the time-translational invariance, i.e. a signal shifted in the time-domain is equally shifted in the wavelet-domain.

With regard to the non-orthogonal case, erasing the redundant information can not be easily performed by down-sampling. That is why Holschneider et al. [153] introduced the à trous scheme, in which, opposed to down-sampling the signal $y(t)$, the filters $H(z)$ and $G(z)$ are up-sampled. This ensures that each decomposition level has length $N$. Moreover, this scheme causes the sub-bands to be time-shift-invariant.

This is a favorable property which was previously used by [7, 36] for ECG-classification and by [100] for QRS-detection. Both used quadratic spline wavelets introduced by Mallat and Hwang [154] for edge detection in two-dimensional images. The high performance of Li et al. [100] R-peak detector based on “the relation between WT’s and singularities in the signal”[100] found by Mallat [155]19, has proven that the different decomposition levels of the quadratic spline wavelet are suitable for characterizing the ECG-signal.

4.4.6 Choosing the right wavelet

Besides favoring either the à trous or Mallat’s scheme, one is left with the choice of the underlying wavelet. There have been great efforts in evaluating different wavelet families and subtypes for QRS-detection and ECG-classification, although no consensus has been found yet.

Choosing wavelets by optimality criteria The search for the optimal wavelet is heavily connected to the chosen criterion / criteria of optimality. The three most common wavelet families:20 Symlets,

19 Based “on the pioneering work of Grossmann [156]”[100] 20 see [157]
Coiflets and Daublets have been developed to address quite different requirements:

- **Symlets**: optimal linear phase response (constant group delay) for an orthonormal wavelet (denoted: sym2, sym3, ... sym(order)\(^{21}\))

- **Coiflets**: symmetric vanishing moments in the time and frequency domain of the wavelet (denoted: coif1, coif2, ... coif(order)\(^{22}\))

- **Daublets**: minimum / maximum phase response (denoted: db1, db2, ... db(order)\(^{23}\))

Still, those criteria do only implicitly account for the experimental data at hand, as the practitioner will choose the wavelet according to the anticipated type of signal. However, those three families represent only a tiny portion of the large number of available wavelets. Therefore, a scheme is needed in which all wavelets can be represented. Moreover, this scheme should allow easy optimization with respect to several criteria. The works of Vaidyanathan [158] and Sherlock and Monro [159] provide just that. The next paragraph will give a short summary of the scheme of [159] which is closely related to the summary given by Froese et al. [70].

### A universal wavelet representation

Let us recall "the transfer function of the low-pass filter in the z-domain" [70] of length \(2N\):

\[
H^N(z) = \sum_{k=0}^{2N-1} h_k z^{-k} = H^N_0(z^2) + z^{-1}H^N_1(z^2)
\] (4.61)

The overall filter bank can be represented by:

\[
F^N(z) = \begin{bmatrix} H^N_0(z) & H^N_1(z) \\ G^N_0(z) & G^N_1(z) \end{bmatrix}
\] (4.62)

Now according to [158] this matrix can be factorized, which yields a recursive formula for obtaining a PR-FB with length \(2N\). The initial step is to choose an angle \(\theta_0\) and to calculate \(C_0 = \cos(\theta_0)\) resp. \(S_0 = \sin(\theta_0)\). All PR-FB of length two have the form:

\[
F^1(z) = \begin{bmatrix} C_0 & S_0 \\ -S_0 & C_0 \end{bmatrix}
\] (4.63)

From this perspective, the Haar wavelet is simply a rotation by \(\theta\) equals 45 degrees, yielding:

\[
F^{Haar}(z) = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}
\] (4.64)

The generation of longer filter banks is carried out by the recursion:

\(21\) number of wavelet coefficients = \(2 \cdot \text{order}\)  
\(22\) number of wavelet coefficients = \(6 \cdot \text{order} - 1\)  
\(23\) number of wavelet coefficients = \(2 \cdot \text{order}\)
where $C_k = \cos(\theta_k)$ and $S_k = \sin(\theta_k)$. In conclusion, each PR-FB forming an orthonormal wavelet is determined by $N$ angles $(\theta_0, \ldots, \theta_{N-1})$. One easily infers that the number of such PR-FBs is infinite. Hence, an optimality criteria which evaluates the usefulness of a certain PR-FB for the ECG-classification task is necessary.

**Optimality by coding performance** This scheme allows the investigation of a different class of optimality criteria: the coding performance. It evaluates how much of the signal’s energy can be described using a meager set of wavelet coefficients. The coding performance of an $M$-staged wavelet decomposition can be assessed by either the coding gain[70, 160]:

$$CG = \frac{1/M \sum_{i=1}^{M} \sigma_i^2}{\sqrt{\prod_{i=1}^{M} \sigma_i^2}} \quad (4.66)$$

or by the compression ratio[70, 159], i.e. the amount of expressed variance of a set of wavelet coefficients. Let $WT_j(\theta, M)$ be the $j^{th}$ wavelet coefficient for the $i^{th}$ heartbeat. The wavelet decomposition is carried out using angles $\theta$ at $M$ decomposition stages. As energy conservation holds, the wavelet coefficients contain as much energy as the signal itself. Let $E_i = \sum_{j=1}^{\Delta_{si}} y^2_{ij}$ be the energy of the $i^{th}$ beat. Moreover, let $WS_{ij}(\theta, M)$ be the wavelet coefficients sorted by their energy (squared value) content in descending order. The compression rate at level $\alpha$, i.e. the number of coefficients needed to retain an energy amount of $\alpha \cdot E_i$ is calculated by:

$$CR_i(\alpha, \theta, M) = \arg \min \sum_{j=1}^{R} \frac{WS_{ij}(\theta, M) E_i}{\geq \alpha} \quad (4.67)$$

The studies carried out by Froese et al. [70] "revealed that wavelet optimization actually improved the results by allowing the neural networks to generalize better". They use a limited dataset (9.8% of available data), thus the superiority of the optimization strategies over standard db6 wavelets might be questionable. The problem of choosing an optimal wavelet arose early after wavelet-research gained more momentum. In the early nineties, Senhadji et al. [135] addressed the four most important questions:

1) what is the appropriate WT to use? 2) what are the most relevant features for efficient encoding of cardiac patterns? 3) what decomposition levels must be retained? 4) does WT improve the recognition process?"[135] Answering question one, they state: "There is no theoretical answer at the moment, and the only technique at our disposal is to compare the results provided by several wavelet families."[135]

Hence, one should take a look at the wavelets used for ECG-classification by other authors.

**Wavelets used in literature** Findings by Acir [61] showed superior performance of DWT using a four staged db2 wavelet analysis over sym2, coif2, db1, db6 as well as over discrete cosine and Fourier transformation 24. This confirms results by Übeyli [130] showing that db2 and db1 are top-performers

24 Güler and Übeyli [131] showed that it is, moreover, superior to coif4, sym10 and sym6
when using a four-staged decomposition: "The smoothing feature of the Daubechies wavelet of order 2 (db2) made it more suitable to detect changes of the ECG-signals." The Daubechies wavelet family is also favored by [161]. They demonstrate that the db4 wavelet outperforms coif4, Morlet, Mexican Hat, Meyer and bi-orthogonal wavelets by means of of classifying AF, VT and VT with a radial basis function neural network.

Besides using wavelets basically constructed on Daubechies work (daublets, symlets and coiflets), the à trous scheme is quite popular. Especially the quadratic spline wavelet introduced by [162] is widely employed for QRS-detection and classification [6, 7, 20, 21, 40, 82, 100, 163]. As Llamendo and Martín [40] put it: "in our application, we are interested in keeping the time accuracy as high as possible (at the expense of redundancy)". The previously described time-shift invariance of the à trous scheme is considered very powerful.

Nonetheless, these approaches are severely limited. The limitation "is that the low-pass and high-pass filters must be chosen a priori and are not adapted to optimally describe the experimental dataset" [70]. For this reason, they employ two methods for optimization of the FIR filter banks previously introduced by [160] and [159]. The next subsection will explain both in more detail.

### 4.4.7 The myth of the perfect wavelet in ECG-classification

The last subsection argues in favor of either using db2 wavelets, quadratic spline wavelets ("literature" criterion) or data optimized wavelets (coding performance criteria). All these selection strategies share the same idea: there is a "perfect" wavelet for the data at hand. Findings of [135] show that WT-coefficients used for good signal reconstruction and for beat classification originate from the same decomposition levels. Moreover, they emphasize that cubic spline wavelets and compactly supported orthogonal ones are equally capable of reconstruction and classification. The relation between good signal reconstruction and performance in beat classification is also supported by findings of [160, 164]. Nonetheless, their datasets are quite limited (one record only in [164]) and only one or two arrhythmias have been considered [135].

The problem is quite more complicated: Even though some authors stick to the db2 choice [114], other authors use different features and a lot more arrhythmias. Hence, they come up with different "optimal wavelets". Prasad and Sahambi [17] use 14 different classes of ECG-beats and choose sym6 as their wavelet, which yields best accuracy. Moreover, the features are of great importance. Engin et al. [67] favor the use of the db11 wavelet, due to employing Shannon-entropy as their feature, for which they report db11 to be superior. If Shannon-entropy is a good feature, than db11 is a good way to go.

Therefore, this thesis has three propositions objecting to the perfect wavelet approach for ECG-classification:

1. the experimental data forms clusters, therefore one wavelet will not be optimal for all clusters
2. the feature at hand, determines the optimal wavelet only for this certain feature
3. variance does not necessarily indicate information content

It is obvious that optimizing the wavelet for the overall data will optimize the filters towards the predominant cluster in the data [29]. Moreover, each cluster (arrhythmia) has its own time-frequency characteristic.
Chapter 4 Feature Extraction

Hence, another wavelet will be optimal. The second proposition states that the choice of the wavelet depends on the feature one wants to extract. Therefore, this thesis investigates different features from different wavelets and combines them to a powerful feature set. The third argument is closely related to the principal component analysis (PCA). It is similar to the wavelet transform, as it forms an orthogonal representation of the data, too. Moreover, it has the highest computation gain achievable with block transforms.[146] In chapter seven those propositions will be evaluated using the real ECG-data.

4.4.8 Wavelets and extracted features

In the experiments carried out to support the propositions of the latter subsection, the following wavelets have been used:

- \( M=4 \) staged: db1-4, db6, db11, haar, sym2-4, sym6, sym11, coif1-4
- \( M=3 \) staged: db2, db6, sym6
- \( M=2 \) staged: db2, db6, sym6

The wavelets above incorporate the three major wavelets db2, db6 and sym6, which are considered high-performing with regard to ECG-classification. The other wavelets serve as test candidates. In the case that those three wavelets are in fact the most informative ones, no features of other wavelets should be part of the final feature set. An extensive analysis of all wavelets is not possible. This would necessitate training the angles \( \theta \) to optimize a certain feature given a certain classification problem. This path is computationally expensive and likely sensitive to over-fitting.

In the list above, the 4 stages wavelets are predominant. This is due to the fact that most of the energy of the QRS complex resides in the 4 to 20Hz frequency band. (see [51]) The wavelet used by [102]

30 as most of the proclaimed wavelets, indicates that the main energy lies in the third and fourth scale of the wavelet decomposition. This is quite obvious taking Mallat’s scheme into consideration: At the first stage, the signal (sampling frequency=360Hz) is split into a low-pass (0-180Hz) and a high-pass (180-360Hz) filtered signal. At the second stage, the low-pass signal is again split into two signals (0-90Hz) and (90-180Hz). For \( M=3 \), the frequency content of those signals is 0-45Hz resp. 45-90Hz. The fourth stage simply splits this the 0-45Hz band into two sub-bands of 0-22.5 and 22.5 - 45Hz. Therefore, the four-staged wavelet analysis contains the low-pass and the high pass filter of the spectrum directly connected to the QRS-complex (4- 20Hz).

Features for discrete wavelet decomposition The features finally extracted are calculated using all available partition functions. (see section 4.2) The governing idea is that the wavelet decomposition is a sort of partitioning transform. Physiologic and static delineation performed partitioning in the time-domain. The wavelet transform now partitions the signal into fragments of changing time and frequency resolution. Each fragment or sub-band can again be analyzed using the partition functions mentioned in section 4.2.

This approach is opposed to approaches by [17, 36, 61], which simply use the wavelet coefficients or a subset of those. According to Senhadji et al. [135], there are mainly two techniques for feature extraction: energy in the sub-bands (abnormal beats have energies concentrated at higher decomposition levels) and the maximal coefficient at each decomposition level (sub-band). Calculating the variance of

30 a quadratic spline wavelet
4.5 Higher order spectra

So far this thesis investigated features in the time and frequency domain. More precisely, the focus of frequency analysis was the power spectrum, which measures the contribution of a single frequency to the power of the signal. The calculation of the power spectrum is handy, as it is simply the Fourier transform of the autocorrelation. (see [165]) Given a time series $x(n)$, the autocorrelation $C_x(k)$ is defined as:

$$c_x^2(k) = E\{x(n)x(n + k)\} \quad (4.68)$$

denoting the expectation value as $E$. The power spectrum $C_x^2(\omega)$ is obtained by applying the Fourier transform to the autocorrelation:
Even though one gains an insight into the energy distribution of the signal over the certain frequencies, "the autocorrelation domain suppresses phase information. An accurate phase reconstruction in the autocorrelation (or power-spectrum) domain can only be achieved if the signal is minimum phase."\[165\] Hence, higher order spectra which retain the phase information in the frequency response are needed. Nikias and Athina \[165\] gave a comprehensive analysis about them. They demonstrated that: (1) if the additive noise is Gaussian and the signal is non-Gaussian, (2) the linear system is non-minimum phase, or (3) the process is nonlinear\[166\], higher order spectra provide additional knowledge about the process at hand. The research carried out by Engin and Demirag \[166\] concluded that the ECG contains non-Gaussian noise and entails non-linearity.

Hence, this section will first state the definition of the higher order spectra and then discuss feature extraction techniques provided in literature. Unfortunately, those methods are quite foggy concerning their details and lack from a deeper analysis of the ECG's higher spectra. Therefore, this thesis carried out an analysis of the bi-spectrum (third order statistics) of the ECG showing great differences between arrhythmias. The results and graphs are provided in the third subsection. The section concludes by describing the features extracted from the two-dimensional spectrum by image segmentation and region statistics.

### 4.5.1 Definition of higher order spectra

The power spectrum was previously derived from the autocorrelation function (second-order statistic). The higher order spectra are therefore derived from their corresponding higher order statistic:

- **Power-spectrum:** \( C_2^2(\omega) = \mathcal{F}\{c_2^2(k)\} \) \hspace{2cm} (4.70)

- **Bi-spectrum:** \( C_3^2(\omega_1, \omega_2) = \mathcal{F}\{c_3^2(k, l)\} \) \hspace{2cm} (4.71)

- **Tri-spectrum:** \( C_4^2(\omega_1, \omega_2, \omega_3) = \mathcal{F}\{c_4^2(k, l, m)\} \) \hspace{2cm} (4.72)

- **n\textsuperscript{th}-order spectrum:** \( C_n^2(\omega_1, \ldots, \omega_{n-1}) = \mathcal{F}\{c_n^2(k_1, \ldots, k_{n-1})\} \) \hspace{2cm} (4.73)

Yet this is quite abstract. For a better understanding, let us discuss the meaning of those statistics using some more familiar statistics. The n-th order moment of a signal \( x(n) \) is:

\[
m^r_n(k_1, \ldots, k_{n-1}) = E\{x(n)x(n + k_1) \cdot \ldots \cdot x(n + k_{n-1})\} \tag{4.75}
\]

Let us take a look at the cumulants of orders up to \( n = 4 \). For ease of calculation, let us suppose the first-order cumulant:

\[
c_1^r = m_1^r = E\{x(n)\} \tag{4.76}
\]
is zero. This states that the series has zero mean, which can be easily achieved for any series via mean removal. Then the, higher order statics $n = 2, 3, 4$ are:

\[ c_2^2(k_1) = m_2^2(k_1) \]  
\[ c_3^3(k_1, k_2) = m_3^3(k_1, k_2) \]  
\[ c_4^4(k_1, k_2, k_3) = m_4^4(k_1, k_2, k_3) - m_2^2(k_1) \cdot m_2^2(k_3 - k_2) - m_2^2(k_2) \cdot m_2^2(k_3 - k_1) - m_2^2(k_3) \cdot m_2^2(k_2 - k_1) \]  

(4.77)  
(4.78)  
(4.79)

The zero-crossings of the cumulants are quite interesting, because they are variance ($n = 2$) the skewness ($n = 3$) and the kurtosis ($n = 4$) of the signal. The cumulants of the third and fourth order can therefore be coined as the generalised skewness or kurtosis of the signal.

**Normalization and bias** In the remainder of this section, I will focus on the third-order cumulant, as it is widely used and one can easily investigate its morphology. In addition, it is much faster to compute, than the fourth or fifth order cumulant. As this thesis regards $c_3^3(k_1, k_2)$ as a "generalized" skewness, further normalization is necessary. Skewness of a zero mean signal $x(n)$ is typically defined as:

\[ \gamma_1 = \frac{E[x(n)x(n)\dot{x}(n)]]}{\sqrt{E[x(n)x(n)]}} \]  

(4.80)

Clearly the sample skewness is biased with respect to the population skewness. Therefore, a more unbiased estimate of the population skewness can be obtained using $\frac{\sqrt{N(N-1)}}{N-2} \gamma_1$. This less biased and normalized skewness can now be easily generalized:

\[ c_3^{\text{norm}}(k_1, k_2) = \frac{E[x(n)x(n-k_1)\dot{x}(n-k_2)]]}{\sqrt{E[x(n)x(n)] \cdot E[x(n-k_1)x(n-k_1)] \cdot E[x(n-k_2)x(n-k_2)]}} \]  

(4.81)

**4.5.2 Standard features on cumulants**

Given a typical segment size $\Delta_{\text{seg}}$, the third and fourth-order cumulants comprise $\Delta_{\text{seg}}^2$ resp. $\Delta_{\text{seg}}^3$ coefficients. The question is, which of these are of significance for the classification of the ECG. Research in the ECG-domain has mainly focused on simple static delineation of the cumulant: Yu and Chen [118] used standard deviation, normalized sum, the number of zero crossings and the symmetry of the third-order cumulant. They apply these measures not directly onto the ECG, but rather on wavelet sub-bands calculated using the à trous scheme. Other authors use only certain values of the cumulant. They choose "five points representing each cumulant (second, third, and fourth-orders) evenly distributed within the range of 30 lags"[117]. Osowski and Linh [117] moreover showed that the maximum values of the fourth and third-order cumulants change considerably between different types of arrhythmia. Another important finding is that the variance within the same class of arrhythmia can be effectively decreased using cumulants.

31 see [165] 32 The same procedure is carried out by deLannoy et al. [42] , and Engin et al. [67] but using 25 lags; [41] use 10 different time lags in the range of 0-180
Despite the analysis of [118], all these approaches treat cumulants as a one-dimensional signal, which they are not. Moreover, it is unknown what these five points mean, maybe they choose $c_3(5,0), c_3(10,0)$ and $c_3(15,0)$ etc., or they only use the diagonal form: $c_3(5,5), c_3(10,10), c_3(15,15)$ and so forth. As the third, fourth and fifth cumulants have dimensionality greater one, it is of great interest, which time lag was used to cut the cumulant. This would be necessary for obtaining a one-dimensional signal.

### 4.5.3 Morphology of the generalized skewness

So far one has no information about the shape of the third-order cumulant or generalized skewness. Let us catch up on that. The graph below depicts the generalized skewness of 100 random instances of ventricular premature contractions in the MIT-BIH Arrhythmia database. The graph shows the standard deviation of $c_{3\text{norm}}(k_1, k_2)$ within those heartbeats, divided by the respective median.

The color mapping shows that one can easily define regions of stable standard deviation, which are separated from each other. The border of each region is defined by:

$$\left| \frac{\sigma}{\mu_{\text{med}}(k_1, k_2)} \right| > 5 \quad (4.82)$$

These patterns are quite different with respect to other arrhythmias (see fig. 4.8). These graphs suggest that each arrhythmia could be identified based on the unique structure of their generalized skewness.
4.5 Higher order spectra

The major challenge here is two-fold: first, one can not explore all instances of a certain arrhythmia, as each cumulant is a three dimensional plot (100,000 plots for each beat in the database). Therefore, measures like standard deviation and median facilitate the analysis. Clearly, one will miss potent patterns within the generalized skewness.

This gets even worse. The pictures in fig. 4.8 are generated using only a portion of the available heartbeats. Hence, those regions, their shape and localization could be an artifact of the sampled instances. However, making use of all heartbeats would pose the risk of over-fitting.

4.5.4 Morphologic features on the third-order cumulant

What features should be extracted from the graphs above? Our major pursuit is to carve out those regions and treat those like the one-dimensional partitions in static delineation. One could employ a quite sophisticated framework for image segmentation to extract those regions. This is beyond the scope of this thesis. In addition, the borders are not sharp with regard to the overall population and including or excluding some tuples \((k_1, k_2)\) from those segments is acceptable. The easiest solution is to convert those graphs to a BMP, mask each region manually and map those bitmaps back into the \((k_1, k_2)\) domain. This procedure has been carried out using Corel Photopaint and some python scripts. They yield regions for each of the 13 arrhythmias. The picture below shows the regions of the generalized skewness before and after the conversion.
Chapter 4 Feature Extraction

Figure 4.9: The result of Photopaint and python scripts (right) applied on the generalized skewness (left)

Measuring orientation and level of skewness  Let us keep in mind that the main task is to identify those arrhythmias by their generalized skewness. The assumption made here is that arrhythmias differ with respect to their level of skewness and the orientation of the skewness in certain areas. It is up to experimentation to show that the inference made by the 100 instances also holds in a larger population. Orientation and level of skewness in the carved regions are assessed employing

- min, max, absolute min, absolute max, min-max
- variance, variance of positive resp. negative skewness values
- number of pos. resp. neg. skewness values (ratios with respect to the region and all \((k_1, k_2)\)-tuples)
- median and mean (both also with respect to negative and positive values)

Gradient at borders and areas  The borders in the pictures above result from areas, where the orientation of the skewness switches between positive and negative skewness. Hence, they mark an area, where there is a strong gradient. The assumption is that the different arrhythmias will show high gradients at exactly those borders. An RBBB beat should show high gradients at the areas carved out for the RBBB beat. The gradient of a one-dimensional signal \(x(n)\) is simply \(x(n) - x(n - 1)\). Calculating the gradient is typically used for high-pass filtering and magnification of fast changes in the signal. The two-dimensional gradient calculation, though, is tightly linked to the notion of edge detection. It makes extensive use of first- and second-order derivatives to extract the edges within images.

The third-order cumulant can be viewed as a simple image which consists of areas of low gradient and borders between those areas, where gradient is expected to be quite high. The basic assumption for classification based on those gradients is this: Let us suppose the shape of the generalized skewness of a certain sample matches with the skewness of the 100 previously selected samples. In this scenario, the gradient in the carved out areas should be low and should be high at the borders. This assumption can be slightly relaxed. If there is a structure within the cumulants of certain arrhythmias, it will manifest in even a small group of 100 heartbeats. Thus, the statistical properties within the carved out regions should be similar with regard to similar arrhythmias. This should also hold for the borders of those regions. If there is no structure within the cumulants of alike classes of heartbeats, the carved out regions will randomly select skewness values. Consequently, the statistical properties will be randomly...
distributed over all classes. Feature selection will show, if either there is structure, or there is none.

### Calculation of gradients

The calculation of the gradients is based on the Sobel-operator (see [167] for further information), which is used for standard edge detection algorithms like the Canny-edge detection. The horizontal and vertical gradients of an $N\times N$ matrix $X$ can be calculated via convolution (denoted as $*$) with a horizontal and vertical gradient kernel ($K_h$ resp. $K_v$):

\[
G_h = K_h \ast X = \begin{bmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1 
\end{bmatrix} \ast X
\]

(4.83)

\[
G_v = K_v \ast X = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
+1 & +2 & +1 
\end{bmatrix} \ast X
\]

(4.84)

The vertical and horizontal gradients are further employed to calculate the overall gradient amplitude $G$ and phase $\theta$:

\[
G = \sqrt{G_h^2 + G_v^2}
\]

(4.85)

\[
\theta = \arctan\frac{G_v}{G_h}
\]

(4.86)

Let us take a look at a nice property of the Sobel operator which eases computation and yields insights about its function. The kernels of the Sobel operator are separable, i.e. they are the outer product of two one-dimensional vectors.

\[
K_h = \begin{bmatrix}
1 \\
2 \\
1 
\end{bmatrix} \begin{bmatrix}
-1 & 0 & 1 
\end{bmatrix}
\]

(4.87)

\[
K_v = \begin{bmatrix}
-1 \\
0 \\
1 
\end{bmatrix} \begin{bmatrix}
1 & 2 & 1 
\end{bmatrix}
\]

(4.88)

The separability of the kernels causes a tremendous computation boost because those two adjacent one-dimensional convolutions are faster than the two-dimensional one. The speed-up for an arbitrary kernel $K$ of size $P \times Q$ is given by $^{33}$:

\[
\frac{P \cdot Q}{P + Q}
\]

(4.89)

$^{33}$ see [168]
Chapter 4 Feature Extraction

The vector \([-1 \ 0 \ 1]\) simply defines the first order gradient in one dimension. The second vector is a weighting vector. The Sobel operator takes into account that the gradient at a certain position is influenced by the gradients in its vicinity. Thus, the horizontal and vertical gradients agglomerate the gradients of the neighboring "pixels" and weight the central pixel twice as strongly. Clearly, this is an arbitrary choice of weights. Operators like the Prewitt-Operator omit weighting entirely.

Using only the very next neighbors is a potential pitfall in the case of the generalized skewness. The borders of the carved out regions are more fuzzy. Hence, the gradients should incorporate more values of neighboring pixels. This will have a low-pass filtering effect which should increase stability of the derived measures.

**Generalization of the Sobel-operator** Let us take a closer look at the weighting vector. This vector is by itself a convolution:

\[
\begin{bmatrix}
1 & 1 \\
1 & 1 \\
\end{bmatrix} * \begin{bmatrix}
1 & 1 \\
1 & 1 \\
\end{bmatrix} = \begin{bmatrix}
1 & 2 & 1 \\
\end{bmatrix}
\]  

(4.90)

In a more general framework, the weighting vector is denoted as, \(w(n = 1)\):

\[
w(n) = e_{n+1} * e_{n+1} | e_{n+1} = [1, \ldots, 1] \in \mathbb{Z}^{n+1}
\]  

(4.91)

In conclusion, the \(w(n)\) denotes a weighting vector, which agglomerates the gradients of \(n\) neighboring pixels. Thus, \(w(10)\) weights the gradients of the 10 neighbors on the right (top) and left (bottom) side of the central pixel.

**Gradient features** The features calculated for the two-dimensional gradient are based on the total amplitude \(G(n)\) and phase \(\Theta(n)\) with regard to the weighting vector \(w(n)\). The phase and amplitude within each carved out region are assessed via the segment functions, described for the skewness itself. Moreover, those functions are employed to evaluate the gradients at the border of those regions. As the curvature of those segments can not be parameterized easily, some basic image processing has to be carried out.

**Border extraction** A single region comprises a set of index tuples \(\hat{R} = \{(h_i, v_i)\}\), which denote pairs of horizontal \(h_i\) and vertical \(v_i\) indices of the generalized skewness. This set can be expressed as a matrix \(R \in \{0, 1\}^{D \times D}\), where \(D\) is the dimension of the generalized skewness:

\[
R_{ij} = \begin{cases} 
1 & \text{if } (i,j) \in \hat{R} \\
0 & \text{else}
\end{cases}
\]  

(4.92)

The matrix \(R\) can be viewed as a digitization of the generalized skewness. If the timelags \((k_i, k_j)\) of the cumulant contribute to the region, \(R_{ij}\) is one. In other cases it is zero. The extraction of the border is carried out by generating a dilated version of the region and subsequent removal of the original version.
4.6 Auto regressive models

The basic pursuit for using auto-regressive models (AR) is that several authors \[61, 67, 90, 166\] came up with promising results employing AR. "The advantage of AR modeling is its simplicity and its suitability for real-time classification at the ICU or ambulatory monitoring." The main assumption of AR modeling is that the sample at time \(t_i\) is a weighted sum of \(P\) earlier samples with \(t < t_i\). The weights are the coefficients of the auto-regressive model of order \(P\). Mathematically spoken the amplitude \(y_i\) can be approximated by:

\[
y_i = \sum_{j=1}^{p} a_j y_{i-j} + \epsilon_i \tag{4.94}
\]

The calculation of the coefficients \(a_j\) can be easily carried out using a least squares approach or the maximum entropy approach by Burg \[169\]. A concise description of the latter method can be found in \[170\]. It will be used as it is most commonly used for AR modeling in ECG-analysis. The source code can either be extracted from \[171\] or as I did from \[172\].

AR modeling is solely to evaluate previous results by other authors using more data. Moreover, it will be interesting to see how this feature-extractor will compete with others, especially as it will use less features than the other ones. Finally, the assessment of AR modeling on a wide range of model orders \(P\) will show if the ECG can be described as a linear feedback loop.

4.7 RR-interval features

At the end, the last sections focused on the morphology of the QRS-complex, ignoring that some arrhythmias do not alter the QRS morphology much. Let us take a closer look at atrial premature contractions (APC): In this case the excitation of the atria is premature and as long the AV-node and other compartments of the electrical conduction system of the heart are not refractory, a normal excitation of the ventricle will occur. This leads to a physiologically shaped QRS-complex, but shifted in time towards the last heartbeat. Distinguishing between normal beats on one hand and premature beats on the other hand necessitates a thorough description of the prematurity of the heartbeats.
4.7.1 Direct measures of prematurity

One standard feature used by several others [20, 21, 30, 41, 58, 71, 99, 117, 118, 141] is the instantaneous RR-interval, i.e. simply the RR-interval of the current heartbeat\textsuperscript{34}. This feature should be shortened in the case of a premature beat, but it is quite unspecific: most premature beats have low RRI, but only few low RRIs are connected to premature beats. Therefore, some authors use the difference between the current interval and the previous interval [20, 21, 38, 68, 83, 118] or the ratio between those intervals [36, 68, 118] as features. Moreover, the difference to the next RR-Interval is used [20, 21, 38, 68, 83]. Another approach would be to use the RR-interval before and after the current RR-interval as a feature, Prasad and Sahambi [17], without any normalization.

The latter features share the same pitfall: measuring the prematurity by the difference or as the ratio between the current and previous resp. next RR-interval will not help in detecting salvos of premature contractions. For this reason, many authors additionally use the average RR-interval previous to the current RR-interval to present their algorithms with a valid normalization. To be more specific, these authors use the average of the last ten beats [30, 41, 117] before or surrounding the ten beats [38, 83] before the current RRI. Park et al. [41] directly normalize the current RR-interval to the mean of the last ten RRIs. Unfortunately, the mean value of the last ten beats can be misleading in the case that the deviations in the RRI signal are too big. The potential drawbacks arising from the usage of the mean value can be compensated if one uses the more robust median value [20] or uses larger time spans [21] (1, 5, 10, 20 minutes before the current RRI). If the ECG is not analyzed in real time, one can also measure prematurity with respect to the average heart rate throughout an ECG-record [35, 38].

The previous remarks simplify the effects of a VPC or APC on the RRI series because they do not only shorten the current RRI with respect to the previous one, but also are followed by a prolonged RRI: the compensatory pause. The pause is caused mainly by VPCs in the following way: During normal sinus rhythm the sinus-node excites the atria from where the excitation is conducted through the whole electrical conduction system of the heart by a widely fixed schedule. The cause for the compensatory pause is that the cells of each compartment need to restore their excitability after the action potential has reached them. Those compartments will be refractory for next 100 to 300 ms. In the case that a VPC occurs, the whole ventricle, the Purkinje fibers and, most important, the AV-node will be refractory for a couple of 100 milliseconds after the VPC. But this is exactly the timespan in which the sinus node excites the atria in its normal cycle. So the excitation can not be conducted from the atria through the AV node in the ventricle because the AV node is refractory. The next excitation and, therefore, the next normal beat will occur in the next cycle of the sinus node. This is the reason why typically the short beat of the VPD plus the compensatory pause match two RR-intervals at normal sinus rhythm. Measuring this short-long structure and characterizing the compensatory pause is therefore a promising tool for the detection of VPCs and APCs. The measure presented so far do focus solely on the short period, i.e. the prematurity. During my diploma thesis I developed a composite index, the detrended squared differenced word filter (DSDW), indicating the existence of a premature beat using so called words. Those can be used to describe several structures, e.g. word 118 is equivalent to short-long-normal, 119 is short-long-long, 120 is short-long-normal. The shortness or longevity of a certain RRI is determined using a threshold $\alpha$. (see Rockstroh et al. [173] for more details)

In conclusion, the following direct measures will be used: Prematurity and Compensatory pause measures ($RR_{-1}/RR_{-1}$, $RR_{1}/RR_{-1}$, $RR_{1}/RR_{0}$, $RR_{0}/RR_{-1}$, $RR_{1}/RR_{1}$), instantaneous RR-interval, average over last 5 and next 13 beats and instant RR-interval ($RR_{0}$, $RR_{5}$, $RR_{13}$), and the current RR-interval with respect to the mean of the last 5 beats ($RR_{0}/RR_{3}$). Furthermore, nine configurations of the DSDW algorithm are used: DSDW looking for word 118 with a threshold of $\alpha \in \{0.01, 0.5, 1., 1.5, 2.\}$ and DSDW searching for words 117-120 with thresholds $\alpha \in \{0.01, 0.5, 1., 1.5\}$.

\textsuperscript{34} [99, 116] use the instantaneous heartrate which is the inverse of the RRI
4.8 The feature database

<table>
<thead>
<tr>
<th>Feature type</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stationary Wavelet</td>
<td>641,144</td>
</tr>
<tr>
<td>Cumulants</td>
<td>437,700</td>
</tr>
<tr>
<td>Static delineation</td>
<td>160,286</td>
</tr>
<tr>
<td>Discrete Wavelets</td>
<td>101,948</td>
</tr>
<tr>
<td>Physiologic delineation</td>
<td>22,836</td>
</tr>
<tr>
<td>AR-Model</td>
<td>4950</td>
</tr>
<tr>
<td>RR-Intervals</td>
<td>24</td>
</tr>
<tr>
<td>All-total</td>
<td>1,368,864</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of the amount of features generated by all feature-extractors.

4.7.2 Indirect measures of prematurity

The last subsection introduced some promising features that directly describe the prematurity of the VPC or APC. Nevertheless, other indirect methods that describe the RRI time series by time domain, frequency domain and non-linear methods exist: namely the heart rate variability (HRV) and the heart rate turbulence (HRT) analysis.

Even though Asl et al. [59] use a variety of standard HRV parameters on 32 RR-intervals surrounding the current RR-interval, HRV analysis is widely ignored in ECG-morphology classification. In the author’s opinion the ignorance is valid because those measures will not alter tremendously from one beat to the other. Furthermore, in the presence of a VPC or APC, those features should be altered approx. for the next 16 beats, which is not sufficient for an exact classification of a single beat.

On the other hand the lack of attention payed to HRT analysis may be caused by the fact that it is a quite new approach and the physiologic foundations are not yet fully understood. The basic principle behind HRT is that a premature beat and its following compensatory pause cause a "turbulence" first in the blood pressure and mediated by the baro-reflex receptors in the aorta, creating a characteristic turbulence in the aftermath of the PVC. The idea behind HRT analysis in PVC detection is that HRT measures should be characteristic only for PVCs and should be nondescript for all other morphologies.

In conclusion, the following HRT features will be used in this thesis: HRTOnset, HRTSlope [174], HRT-Timing [175], HRTJump, HRTDurationLength, HRTNTP [176].

4.8 The feature database

A major concern of this thesis is the comparison of feature-extractors and finding a suitable way to select features from a large amount of features. The feature-extractors generate around 1.4 million features. This section is intended to provide the reader with a rough summary of the feature-extractors and the amount of features they generate. Table 4.1 shows the basic statistics. More detailed information can be found in the respective paragraphs.

---

35 frequency domain (cubic spline 4Hz interpolation, LF/HF reflecting autonomic balance between vagal and sympathetic activity), time-domain (Mean, RMSSD, SDNN, SDSD, pNN50, pNN10, pNN5), non-linear measures (Poincaré SD1, SD2, Approximate Entropy, Spectral entropy, Lyapunov exponent, Detrended fluctuation analysis (DFA), sequential trend analysis) 36 never mentioned in any publication screened in this thesis
Auto regressive models  The auto regressive model of order $P$ and its respective coefficients $a_j$ are calculated using the maximum entropy approach by Burg [169]. (see section 4.6) The order (the number of coefficients) of the model remains fuzzy and can not be determined precisely. Hence, $P \in \{1, \ldots, 100\}$. One can easily infer that in sum $100 \times 99/2 = 4950$ AR coefficients are calculated.

Physiologic delineation  The first properties calculated are the timings yielded by the ECGpuwave algorithm: P-onset, P-offset, P-center, QRS-onset, QRS-offset, QRS-center, T-onset, T-offset, T-center1, T-center2, T-type1, T-type2. The types are integers denoting the shape of the first or even second T-wave. Each of the 10 timings were calculated relative to QRS-center, i.e. the position of the R-peak. Moreover, the amplitudes at each position have been determined.

The ECG was partitioned using asymmetric partitioning $p^a$ employing the timings of the PQRST points. For instance, the segment between P-onset and P-center was extracted to analyze the shape of the P-wave. For each segment the 22 major segment functions have been calculated, to ensure a better comparability between physiologic and static delineation of the ECG, which makes extensive use of those partition functions. Moreover, the duration of each partition was measured. Thus, overall 23 partition functions have been applied to 15 partitions. An exact description of the asymmetric partitions and the partition functions is given in section 3.2. For each function the relational feature with regard to the 15 segments has been calculated using $\rho^\text{Div}$ and $\rho^\text{Sub}$. This adds another $1.5(N^2 - N) = 315$ features per partition function (N = 15 partitions).

<table>
<thead>
<tr>
<th>Feature type</th>
<th>remarks</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRS-timings</td>
<td>Positions of PQRST on and offset</td>
<td>10</td>
</tr>
<tr>
<td>QRS-amplitudes</td>
<td>ECG-amplitudes at those positions</td>
<td>10</td>
</tr>
<tr>
<td>QRS-annotations</td>
<td>Two T-wave annotations</td>
<td>2</td>
</tr>
<tr>
<td>QRS-partitions</td>
<td>23 partition functions on 15 partition</td>
<td>345</td>
</tr>
<tr>
<td>QRS-relational</td>
<td>315 relations performed for each of 23 partition functions</td>
<td>7245</td>
</tr>
<tr>
<td>QRS-features</td>
<td></td>
<td>7612</td>
</tr>
<tr>
<td>QRS-total</td>
<td>All three normalizations on a single 400 samples segment</td>
<td>22836</td>
</tr>
</tbody>
</table>

Table 4.2: Summary of the features used in this thesis related to the physiologic delineation of the QRS-complex: The number of relations per partition function is given by $1.5(N^2 - N) = 315$ employing $\rho^\text{Sub}$ and $\rho^\text{Div}$.

Static delineation  Two different segment sizes have been analyzed employing all three normalization modes and also searched one in between segment using MinMax normalization. All 22 partition functions have been evaluated on partitions of size: 3, 5, 8, 16. This is intended to catch features that emphasize either on the center or the edges of the QRS-complex. Moreover, the relations between those partitions were computed on a per function basis.

<table>
<thead>
<tr>
<th>Feature type</th>
<th>calculation</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base static</td>
<td>22 effect funcs on 32 partitions</td>
<td>704</td>
</tr>
<tr>
<td>Base static rel.</td>
<td>1488 relations performed for each of 23 partition functions</td>
<td>21824</td>
</tr>
</tbody>
</table>

Table 4.3: Summary of the features used in this thesis related to the static delineation of the QRS-complex: The number of relations per partition function is given by $1.5(32^2 - 32) = 1488$ employing $\rho^\text{Sub}$ and $\rho^\text{Div}$.

Not only the segments statistics have been evaluated. In total, 2590 features measure the width and asymmetry of the QRS-complex. Hence, the number of zero-crossings $Z_i(\alpha)$ and the total width of the QRS-complex $Z_w(\alpha)$ (eq. 4.30 and eq. 4.33 have been calculated using $\alpha \in \{0.9, 0.75, 0.5, 0.25, 0.1\}$. This also holds for the left and right asymmetry (eq. 4.32 and eq. 4.31).
4.8 The feature database

<table>
<thead>
<tr>
<th>Feature type</th>
<th>calculation</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Wi</td>
<td>( Z_w(\alpha) ) and ( Z_i(\alpha) ) for 5 heights (( \alpha )) for two normalizations</td>
<td>20</td>
</tr>
<tr>
<td>Base Wi rel</td>
<td>Relational features for each normalization ( 1.5(5^2 - 5) )</td>
<td>60</td>
</tr>
<tr>
<td>Base Asy</td>
<td>( Z_l(\alpha) ) and ( Z_r(\alpha) ) for 5 heights (( \alpha )) for two normalizations</td>
<td>20</td>
</tr>
<tr>
<td>Base Asy rel</td>
<td>Relational features for each normalization ( 1.5(10^2 - 10) )</td>
<td>270</td>
</tr>
<tr>
<td>Final Wi+Asy</td>
<td></td>
<td>370</td>
</tr>
<tr>
<td>Final Static</td>
<td>( 370 + 704 + 21824 )</td>
<td>22898</td>
</tr>
<tr>
<td>Total Static</td>
<td>3 norms * 2 segments + 1 segment</td>
<td>160286</td>
</tr>
</tbody>
</table>

Table 4.4: Summary of the features used in this thesis related to the static delineation with regard to symmetric and asymmetric measures of the QRS complex width.

**Stationary wavelet transform**  In the case of the Stationary Wavelet Transform, one makes use of all features for static delineation. The only difference is that the SWT is used to perform bandpass filtering of the raw signal in advance. At each of the four stages of the SWT a full static delineation is employed. This is only successful because the SWT does not shift the signal and produces a signal which is as long as the raw signal. Hence, SWT produces four times the amount of features of the static delineation: \( 160286 * 4 = 641144 \). The 3-dB bandwidth of the resulting equivalent band-pass filter was calculated according to Li et al. [100] using 360Hz instead of 250Hz.

<table>
<thead>
<tr>
<th>Stage of SWT</th>
<th>3-dB bandwidth in Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWT(1)</td>
<td>90 - 180</td>
</tr>
<tr>
<td>SWT(2)</td>
<td>26 - 84</td>
</tr>
<tr>
<td>SWT(3)</td>
<td>12 - 39</td>
</tr>
<tr>
<td>SWT(4)</td>
<td>6 - 19</td>
</tr>
</tbody>
</table>

Table 4.5: Frequency content in the sub-band extracted at each stage of the SWT

**RR**  There are 24 RR-interval measures used in this thesis. They are summarized on page 77.

**Discrete Wavelets**  First of all we calculated the standard wavelets: two and three staged: db2, db6, sym6; four staged: Haar, db1-4, db6, db11, sym2-4, sym6, sym11, coif1-4. For each of the 22 wavelets the 22 major partition functions have been computed for each sub-band. Hence, each wavelet yields \( (\text{stage} + 1) \cdot 22 \) features which sums up to 2222 features based on 101 sub-bands. For each wavelet the sub-bands have been exploited to derive relational features. Thus, each wavelet contributes \( 1.5 \cdot \text{stage} (\text{stage} + 1) \) features employing the \( \rho_{\text{Sub}} \) and \( \rho_{\text{Div}} \) relation. All those wavelet features are calculated on the three ECG-segments: two symmetric around the R-peak and one in between the R-peaks of adjacent heartbeats. The two symmetric segments have been normalized using three different normalization techniques described in section 3.4. Table 4.6 summarizes the features used within this thesis with regard to discrete wavelet decomposition.

<table>
<thead>
<tr>
<th>Feature type</th>
<th>remarks</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet-base</td>
<td>22 partition function times 101 sub-bands</td>
<td>2222</td>
</tr>
<tr>
<td>Wavelet-relational</td>
<td>561 relations performed for each of 22 partition functions</td>
<td>12342</td>
</tr>
<tr>
<td>Wavelet-features</td>
<td></td>
<td>14564</td>
</tr>
<tr>
<td>Wavelet-total</td>
<td>All features calculated on the 7 standard segments</td>
<td>101948</td>
</tr>
</tbody>
</table>

Table 4.6: Summary of the features used in this thesis related to discrete wavelet decomposition (DWT): The number of overall analyzed sub-bands is \( 3 \cdot (3 + 4) + 3 \cdot 16 = 101 \) and the number of relations per partition function is \( 1.5 \cdot [3 \cdot (3 \cdot 2 + 4 \cdot 3) + 16(5 \cdot 4)] = 561 \) employing \( \rho_{\text{Sub}} \) and \( \rho_{\text{Div}} \).
Table 4.7: Summary of the features used in this thesis related to higher order spectras, namely the third order cumulant.

<table>
<thead>
<tr>
<th>Feature type</th>
<th>remarks</th>
<th># of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumulant-base</td>
<td>20 evaluators on 185 sectors</td>
<td>3700</td>
</tr>
<tr>
<td>Cumulant-rel.</td>
<td>relative values per shape (1.5(N^2 - N)) for each evaluator ((3260 \times 20))</td>
<td>97800</td>
</tr>
<tr>
<td>Cumulant-gradients</td>
<td>Sobel-operator with 3 smoothing levels, 20 evaluators on each of 185 sectors ((\text{amp, phase, ampbor, phasebor})) (20 \times 185 \times 4 \times 3) = 44400</td>
<td></td>
</tr>
<tr>
<td>Cumulant-features</td>
<td></td>
<td>145900</td>
</tr>
<tr>
<td>Cumulant-total</td>
<td>3 normalizations</td>
<td>437700</td>
</tr>
</tbody>
</table>

**Cumulants** According to section 4.5 this thesis investigated on third order cumulants. More precisely it identified regions of low variability and potentially high information content across 12 arrhythmias. For each arrhythmia those regions or segments have been evaluated using 20 sector functions:

- Minimum, Maximum, Max - Min
- Absolute Maximum and Minimum
- Variance, pos. Variance, neg. Variance
- number of positive / negative cumulants
- ratio of positive / negative cumulants in the given sector
- ratio of positive / negative cumulants with regard to all cumulants
- median, median of positive / negative cumulants
- mean, mean of positive / negative cumulants

In total, this thesis analyzes 185 sectors derived from several arrhythmias: FP(17 segments), Fuse (12), LBBB (12), N(13), NESC (10), NPREM (13), RBBB (23), SVES(17), VES (17), VESC(7), vP(5), Stat7 (39). Stat7 does not denote a certain type of heartbeat. It denotes splitting the third order cumulant into 39 in partially overlapping sectors. For more details see section 4.5. For each of those 13 sector-groups, relational features have been calculated as previously described in the wavelet and physiologic delineation case. Besides the evaluation of the sectors alone, the gradients at the borders of the identified regions (ampbor, phasebor) have been computed as well as within those regions (amplitude and phase). The amplitude and phase of the gradients has been assessed using a standard Sobel-operator with smoothing. Smoothness was varied between \((2, 6, 10)\). For each smoothing level the amplitudes and phases of the gradients have been calculated. They have been evaluated using the above sector functions on all 185 sectors and the borders around those sectors.
Chapter 5

Feature Selection

The last chapter explained the extraction of features from the preprocessed signal by wavelet transformation, physiologic or static delineation and other methods. In total, 1.4 million features have been created. Hence, one faces two major obstacles for applying elaborate methods of pattern recognition. First, Support vector machines (SVM), Artificial Neural Networks (ANN) and others can hardly cope with this amount of data. This is due to high dimensional feature sets being computationally intractable by means of memory consumption and computation time during training. Second, the calculation of that many features for every heartbeat is neither feasible nor reasonable.

Hence, a reduction of features is necessary. This procedure is also called dimension reduction. There are several ways this can be carried out. Principal Component Analysis (PCA) and Clustering are strong candidates. Clustering algorithms seek to find structures in the high-dimensional space, e.g. spheres or ellipsoids containing large amounts of data which are separate from other spheres. After the parameters of those structures have been determined, each point is represented by its relation to this structure, e.g. the distance to the center of the centroid or the membership or probability of membership in one of the spheres.

PCA, on the other hand, searches for those linear combinations of features which contain the highest amount of energy (variance) with respect to the data. Obviously, both methods can reduce dimensions tremendously. However, those methods do not get rid of the second obstacle. One still has to calculate 1.4 million features for each heartbeat in order to obtain the linear feature combinations, high-dimensional distances and sphere-memberships.\(^1\)

In consequence, another branch of dimension reduction seems more reasonable: feature selection. In fact, any automated technique for selecting a smaller feature set from a larger one can be considered as feature selection. Hence, thousands of feature selection techniques have been proposed over the last few decades.

That is why the first section of this chapter is devoted to the taxonomy of feature selection methods. Why filtering methods are the most suitable ones for the problem at hand, i.e. selecting 20-60 features out of 1.4 million in a multi-class (C=13) environment, will be explained.

The second section is dedicated to feature ranking for binary or two-class problems. Feature ranking is the most important task for a feature filtering method as it measures the importance of a certain feature. This section will present current feature-ranking techniques and their connection to effect-size indices. It derives new ranking methods originating from text-categorization. The latter focuses on ranking strategies which are applied on dichotomized features. However, features derived from the ECG are of cardinal scale which would necessitate a dichotomization and a technique to extract a final ranking

\(^1\) References to authors employing PCA for ECG-classification: [23, 120, 138, 163, 177]
score. This thesis provides a strategy to accomplish this task and will demonstrate the superiority of the dichotomized effect-size indices (DESI) over standard ranking methods in the seventh chapter.

The third section is concerned with the question of extending the binary-class ranking methods to a multi-class situation. It provides some general remarks on multi-class feature ranking, common methods employed for this purpose and a general, but computationally intense, approach for doing so. The fourth section explains several novel methods for performing fast and high-performing multi-class feature selection.

5.1 The taxonomy of feature selection

There are four major feature selection approaches: exhaustive search, filters, wrappers and embedded methods. [178] Clearly, this subdivision originates from a technical perspective. It reflects basic methodological differences between those feature selection strategies. In brief: exhaustive search explores the classification power of all possible feature sets; filters rate features by statistics with no regard to the classifier to be used later on; wrappers rank features according to performance measures calculated by the classifier in question and embedded methods are machine learning methods in which feature selection is inherent. This is surely a viable subdivision.

Another taxonomy by [179], on the other hand, is more focused on the structure of the resulting feature set itself. They distinguish between optimal (exhaustive search, branch-and-bound) and suboptimal feature selection. The suboptimal methods again branch into techniques either providing a single or multiple solutions. Moreover, one could differentiate between deterministic and static strategies. With respect to that taxonomy, this thesis will focus on feature selection techniques generating a single solution in a deterministic, suboptimal but fast way.

5.1.1 Exhaustive search

The most natural way to solve the feature selection problem is to try all possible feature subsets, measure their classification performance and opt for the best subset. But obviously, there is a major setback: "The computational requirements of exhaustive search methods (those which test all possible subsets) increase exponentially with the number of features on the original set." [7]. More precisely, they increase with $2^F$, given the number of features $F$ in the input feature set. The so called curse of dimensionality renders exhaustive search "impracticable for sets with over a dozen or so features"[7].

5.1.2 Wrapper methods

All wrapper methods share the same basic idea: Generate a start feature subset from the overall feature set and calculate the performance of the classifier (SVM, NN, HMM). Then, remove or add a certain amount of features and check the classification performance again. The wrapper method at hand is driven by the change in performance which causes the algorithm to shrink or extend the feature set. Methods called forward wrappers start with a small basic feature set and add features to increase the classifier's performance. Backward wrappers, on the other hand, start from the maximum feature set
and remove features from it to increase the performance. Of course, there are hybrid methods that use a combination of both approaches. 2

The major advantage of wrapper methods is that they use "some kind of performance measure based on the results of testing the candidate subsets with the classifier itself."[7] So, the computational expense is compensated by the fact that wrapper methods account for the "structural characteristics imposed by the classifier"[7]. Unfortunately, wrappers are rendered impracticable in the situation at hand 3 for three reasons:

- time consuming calculation
- low impact of a given feature in high dimensional feature space
- a preliminary filter method is needed

As previously stated, a wrapper method needs to recalculate the classifier's performance in each step. If one uses large amounts of training data and tries to cross-validate those results, each step will take at least one hour for SVM calculation with a typical test configuration. 4 Let us suppose one wants to search for a subset of 10 features out of 1.4 million. A typical forward wrapper would need approximately 14 million runs of a multi-class SVM. Surely, this amount can be reduced in case some order of the features can be imposed. Nonetheless, even reducing this procedure to around 1000 steps would still exceed the computational capabilities available. Chang and Lin [181] state with regard to a dataset (named SIDO) comprising “only” 5000 features, that leaving one feature out methods (a backward wrapper method) "are infeasible for SIDO due to the large number of features of SIDO."

The second argument is probably the most important one: The SVM tries to optimize the distance of all input vectors from the parting hyper-plane. Therefore, the SVM needs a metric to measure the distance between two points in the feature space. The majority of metrics share one drawback. Distances between two points in feature space will homogenize if the dimension of the feature space increases. More precisely, the impact \( r \) of a change in one component of the feature vector will diminish if the dimension of feature space is high enough. 5 This can be shown using simplified assumptions of the feature space of size \( N \).

Given the Euclidean metric:

\[
\|x - y\| = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}
\]  

the impact \( r \) of the j-th component changing by a certain factor \( \alpha \in \mathbb{R} > 0 \) on the total distance between two points \( x \) and \( y \) is:

\[ \text{For example the Sequential Floating Forward Selection (SFFS)[180] 3 1,400,000 highly correlated features 4 Referring to tab. 7.8, one can see that calculating the ranking score of all features (5 minutes) is around 10 times faster than even a single SVM assessment of one subset 5 In a physics picture, this is congruent with the fact that the volume of the phase space in thermodynamics is concentrated in the shell of the phase space and that deviations in microscopic states (features) do not account for the overall state of the system.} \]
Chapter 5 Feature Selection

\[
\begin{align*}
\mathbf{r} &= \frac{\left(\begin{array}{c}
\mathbf{x}_1 \\
\vdots \\
\sqrt{\alpha} \mathbf{x}_j \\
\vdots \\
\mathbf{x}_N \\
\end{array}\right) - \left(\begin{array}{c}
\mathbf{y}_1 \\
\vdots \\
\sqrt{\alpha} \mathbf{y}_j \\
\vdots \\
\mathbf{y}_N \\
\end{array}\right)}{||\mathbf{x} - \mathbf{y}||} = \sqrt{\frac{\sum_{i \neq j} (x_i - y_i)^2 + \alpha(x_j - y_j)^2}{\sum_i (x_i - y_i)^2}} \\
\end{align*}
\] (5.2)

Introducing the mean distance between components \( |x_i - y_i| \approx \Delta \forall i \) simplifies eq. 5.2:

\[
\mathbf{r} = \sqrt{\frac{(N-1)\Delta^2 + \alpha \Delta^2}{N \cdot \Delta^2}} = \sqrt{1 + \frac{\alpha - 1}{N}}
\] (5.3)

The first astonishing lesson from eq. 5.3 is that, given the presumptions above, the mean feature distance \( \Delta \) is not important at all because \( \alpha \) and \( \mathbf{r} \) scale up in case the mean feature distance is high. In case one removes the \( j^{th} \) feature completely, i.e. \( \alpha = 0 \), the distance will change by \( \mathbf{r} = \sqrt{1 - \frac{1}{N}} \), which simply states that for \( N \rightarrow \infty \) the change \( \mathbf{r} \approx 1 \) is negligible. Let us look at the other extreme. How big should \( \alpha \) be, in order for the feature set to be dominated by the \( j^{th} \) feature? Equation 5.3 states that the mean distance of the \( j^{th} \) feature has to be \( 3N \) times higher as the mean feature distance to double the mean distance between \( \mathbf{x} \) and \( \mathbf{y} \).

The third argument is simply a consequence of the previous two. A backward wrapper would employ the maximal feature set and try to remove features from the feature set. This approach is rendered futile if the impact of one feature is minimal. A forward wrapper would need a minimal set of features that can be grown. Choosing arbitrary features from a set of 1,400,000 features will lead nowhere. Therefore, both wrapper methods and their hybrid versions have to facilitate a preliminary filter method. As Mar et al. [7] puts it: "However, if all features found in the literature were introduced, the number will be indeed too large, rendering a complete FS to be computationally unfeasible, so a constraint of some type was necessary to keep the number within reasonable limits". The primal assumption of this thesis is that this is possible, only if a multi-class feature-filtering method can be developed which is as well computationally cheap and high-performing.

In conclusion, this thesis will use a wide range of potential features and will find a feasible solution to do the filtering automatically. This is in contrast to the feature filtering by mere reasoning performed by Mar et al. [7]. After feature-filtering is performed, one can subsequently employ sophisticated wrappers like sequential floating forward selection (SFFS) introduced by [180]. SFFS has been successfully used by [7, 21, 40]. It demonstrated superior power (see [182]) over less sophisticated approaches like SVM recursive feature elimination (see [141]). Moreover, it is considered to be nearly as good as optimal methods like branch and bound, while being much faster. (see [179]) A major problem\(^6\) shown in [182] is that even though SFFS provides slightly better results than the filter method used in their study, the feature set is much more unstable by means of the Average Tanimoto Index (ATI). It was developed by Kalousis et al. [184] who based the ATI on the notion of similarity measured by the Tanimoto Index. (see [185])

\(^6\) Spence and Sajda [183] points out that SFFS has some problems when facing noisy and small datasets
5.2 Feature ranking by inference testing

\[ ATI(S) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \]  

(5.4)

The ATI measures the average similarity by feature sets computed on different boot strap sets of the underlying data. This definition will be used in chapter six to measure the stability of the feature selection methods developed in this thesis.

**Embedded methods** Embedded methods are different from wrapper methods with regard to how feature selection and machine learning methods are interlaced. A typical wrapper clearly separates feature selection and machine learning. The latter only gives performance measures which guide the feature selection process. In embedded methods, though, the feature selection is inherent in the machine learning strategy. A typical example is pruning in neural networks. Consider 1.4 million input nodes, a hidden layer of a certain amount of neural nodes and an output layer of 13 nodes. This neural network setup has to be calculated and pruning has to disconnect irrelevant input nodes. Still, this procedure is quite time- and memory-consuming. Therefore, embedded methods were not considered in this thesis. For closure, the following author’s made use of embedded feature selection methods for ECG-classification: [59, 63, 118, 181].

5.1.3 Filter methods

A far more appealing approach is to calculate a performance score, i.e. a score calculated on data of the single feature and to rate the discriminant power of the single feature. Afterwards, one can select the best twenty, fifty or more features for the classification.

The NIPS challenge “outlined the power of filter methods. For many years, filter methods have dominated feature selection for computational reasons. ... . It was understood that wrapper and embedded methods are more powerful, but too computationally expensive. Some of the top ranking entries use one or several filters as their only selection strategy.” [186] The following section will give detailed information about current binary ranking strategies.

5.2 Feature ranking by inference testing

The last section showed that filtering is the only feasible solution from a computational point of view, when facing \( F = 1.4e6 \) features. At least to arrive at an amount of 100 to 1000 features that could be used for more sophisticated feature selection approaches. The whole filtering process depends on the quality and reliability of the feature ranking method to assess the relevance of a certain feature. From an inference testing view, relevance and statistical significance are highly correlated. Inference testing assesses the statistical significance of a certain treatment in the medical domain, for example. Using the statistical significance as a feature ranking tool, i.e. to rank each feature by the probability that the

\( ^7 \) An intersection of filtering and wrapper methods are the leaving one feature out approaches. They can be considered fast backward wrappers, as the performance score is the change in computation power in case a feature is removed from the overall feature set. It is fast because the backward wrapper is only used once and afterwards the best ten features are selected. Still, this approach shares the same drawbacks with other wrapper methods concerning the impact of a single feature on the inter-point metrics.
difference between class one and class two is due to chance, is problematic. The statistical significance is a measure that takes into account the sample sizes as well as the difference between the classes' feature distributions. Effect-size indices on the other hand do assess the difference only.

This section will first discuss how standard effect-size indices are linked to inference testing measures. Numerical experiments, discussed in chapter 7, reveal that standard effect-size measures, t-Test and F-test and the Pearson correlation are interchangeable by means of ranking features. Moreover, it can be demonstrated that the area under curve (AUC) of the receiver-operating characteristic (ROC) is the effect-size index related to the Wilcoxon-Mann-Whitney U statistic and is interchangeable with the Spearman rank correlation.

5.2.1 Parametric effect-size indices

The majority of common effect-size indices seek to measure the standardized mean difference (see [187]):

\[ \delta = \frac{\mu_1 - \mu_2}{\sigma} \]  

(5.5)

Let \( y_1 \) and \( y_2 \) denote \( N_1 \) resp. \( N_2 \) samples of a certain feature \( y \) sampled from class one resp. class two. Given the general function:

\[ f(\alpha) = (N_1 + N_2 - \alpha) \cdot \frac{|\hat{y}_1 - \hat{y}_2|}{\left(N_1 \cdot \sigma^2_N(y_1) + N_2 \cdot \sigma^2_N(y_2)\right)} \]  

(5.6)

Cohen’s D \( CD \), Cohen’s D with pooled standard deviation \( CDP \), Hedge’s G \( HG \) and Hedge’s unbiased G \( HGU \) value can be calculated as:

\[ CD = \frac{|\hat{y}_1 - \hat{y}_2|}{\sigma_N(y_1)} \]  

(5.7)

\[ CDP = f(0) \]  

(5.8)

\[ HG = f(2) \]  

(5.9)

\[ HGU = HG \cdot \left(1 - \frac{3}{4 \cdot (N_1 + N_2) - 9}\right) \]  

(5.10)

The latter parameters all share the same idea of effect-size testing. They use only the distance between the sample mean values and divide it by a weighted or unweighted variance. The Fisher score, on the contrary uses another approach by weighting the sum of the squared mean values and keeping the variance unweighted, as can be easily seen in the following equations:

The traditional definition of the Fisher-Score is:
5.2 Feature ranking by inference testing

\[
FS = \frac{(\hat{y}_1 - \hat{y})^2 + (\hat{y}_2 - \hat{y})^2}{\sigma_{N_1}^2(y_1) + \sigma_{N_2}^2(y_2)}
\]  
(5.11)

Exploiting the fact that the pooled mean value \( \hat{y} = \frac{N_1 \hat{y}_1 + N_2 \hat{y}_2}{N_1 + N_2} \), one can simplify equation 5.11:

\[
FS = \frac{N_1^2 + N_2^2}{(N_1 + N_2)^2} \cdot \frac{(\hat{y}_1 - \hat{y}_2)^2}{\sigma_{N_1}^2(y_1) + \sigma_{N_2}^2(y_2)}
\]  
(5.12)

Despite weighting variance or mean values, all these parameters share the same problem: They make assumptions about the underlying distribution. This is a crucial point when testing dozens or thousands of features in an automated fashion. One can not guarantee that all features are normally distributed. Moreover, the sample mean and sample variance heavily depend on the absence of outliers, which would either obfuscate or feign strong features. Besides the problem of outliers, sample mean and sample variance are no proper descriptive variables for assessing the shape of an arbitrary distribution.

The drawbacks of parametric effect-size assessment are inherited from the inference testing methodologies they are derived from. According to \[188\] and \[189\], all measures above can be easily deduced from Student's t-Test, the F-Test and other parametric inference testing techniques. Furr \[190\] and Thalheimer and Cook \[191\] take advantage of this link to estimate the effect-size of medical and psychological outcomes by using the given t- and F-test values. ⁸

5.2.2 Non-parametric effect-size indices

So what could be a potential non-parametric effect-size index? As explained in the last subsection, inference testing and effect-size measures are linked. A potent candidate for non-parametric inference testing is the Wilcoxon-Mann-Whitney U statistic, extensively used in the literature.

It is defined according to eq. 5.13 and employs the rank \( r_i \) of the samples in \( y_{1i} \) with regard to the pooled samples \( y = y_1 + y_2 \).

\[
U = \sum_{i=1}^{N_1} r_i - \frac{N_1(N_1 + 1)}{2}
\]  
(5.13)

Mason and Graham \[193\] summarize the results of \[194, 195\], showing that the U-statistic is linked to the area under curve (AUC) of the receiver-operating characteristic (ROC) by:

\[
AUC = 1 - \frac{1}{N_1N_2} U
\]  
(5.14)

This is a remarkable result because the ROC from which the AUC was basically derived is bypassed. Even though calculating the AUC using the above formula is computationally faster, investigating the

⁸ According to \[192\], there is a lack of mention of effect-size measures, accompanied with measures of statistical significance. One can use the findings above to reverse engineer the effect-sizes.
Chapter 5 Feature Selection

ROC opens the door to a much more vast and diverse class of powerful effect-size measures which are primarily used for feature selection in text categorization.

The ROC can be obtained in the following way:

Let $\epsilon$ denote a certain threshold in the range of $\min(y_1, y_2)$ to $\max(y_1, y_2)$. Without loss of generality, one can declare $\text{median}(y_1) < \text{median}(y_2)$, which simply states that the distribution of $y_1$ is left from the one of $y_2$. With those presumptions, one can easily derive a 2x2 - confusion matrix for every given threshold $\epsilon$:

$$
\begin{bmatrix}
    a(\epsilon) & b(\epsilon) \\
    c(\epsilon) & d(\epsilon)
\end{bmatrix}
= \begin{bmatrix}
    \sum_{i=1}^{N_1} \Theta(\epsilon - y_{1i}) & \sum_{i=1}^{N_2} \Theta(\epsilon - y_{2i}) \\
    \sum_{i=1}^{N_1} \Theta(y_{1i} - \epsilon) & \sum_{i=1}^{N_2} \Theta(y_{2i} - \epsilon)
\end{bmatrix}
$$

(5.15)

With eq. 5.15 at hand, one can deduce the most characteristic measures:

<table>
<thead>
<tr>
<th>measure</th>
<th>equation</th>
<th>measure</th>
<th>equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>members of class 1</td>
<td>$N_1 = a(\epsilon) + c(\epsilon)$</td>
<td>members of class 2</td>
<td>$N_2 = b(\epsilon) + d(\epsilon)$</td>
</tr>
<tr>
<td>class 1 detected</td>
<td>$A_1(\epsilon) = a(\epsilon) + b(\epsilon)$</td>
<td>class 2 detected</td>
<td>$A_2(\epsilon) = c(\epsilon) + d(\epsilon)$</td>
</tr>
<tr>
<td>sensitivity</td>
<td>$T(\epsilon) = a(\epsilon)/N_1$</td>
<td>pos. predictive value</td>
<td>$\text{PPV}(\epsilon) = a(\epsilon)/A_1(\epsilon)$</td>
</tr>
<tr>
<td>specificity</td>
<td>$S(\epsilon) = b(\epsilon)/N_2$</td>
<td>neg. predictive value</td>
<td>$\text{NPV}(\epsilon) = b(\epsilon)/A_2(\epsilon)$</td>
</tr>
</tbody>
</table>

Table 5.1: Characteristic measures of the 2x2 confusion matrix

The threshold $\epsilon$ is not chosen arbitrarily. Clearly, the only points where either $a(\epsilon)$, $b(\epsilon)$, $c(\epsilon)$, $d(\epsilon)$ eventually change are the samples in the pooled data $y$. Consequently, the smoothness of $a(\epsilon)$ is determined by the number of samples in $y$, which is equal to the number of tested thresholds.

The ROC can be obtained by plotting $T(\epsilon)$ against $1 - S(\epsilon)$ as can be seen in fig. 5.1. The curve describes the trade-off between increasing sensitivity (moving to the right) and decrease in specificity. A good discriminator allows a minimum loss in specificity while guaranteeing high sensitivity. The perfect discriminator ($T=1, S=1$) would have rectangular shape, filling the whole box. That is why the AUC, the area under curve, is widely employed to test discriminant power and, as can be seen from the above derivations, is the effect-size corresponding to the Wilcoxon-Mann-Whitney statistic.

### 5.2.3 Dichotomized effect-size indices

Obviously, the ROC is just an example of a much more diverse class of similar testing procedures I would like to call dichotomized effect-size indices (DESI). They have been developed during this thesis and inspired by approaches in clinical trials and text categorization. In clinical trials, dichotomous outcomes are common: For the sake of the argument, say one investigates the survival rate of men with a systolic blood pressure above 120 mmHg and those below. The true positive ones are all men dying within 10 years with blood pressure above 120 mmHg and the true negatives are those surviving with a blood pressure below 120 mmHg. The resulting confusion matrix can be assessed in various ways. In text categorization one faces a quite similar task. One strives to determine which word has to be in the text in order for the text to fall in a certain category. So, the dichotomous criterion is whether or not the word is in the text.
5.2 Feature ranking by inference testing

In both scenarios, the threshold is either fixed a priori (clinical trial) or the outcome is by itself dichotomous (text categorization). The DESI now exploit the concepts developed therein and apply them on cardinal data. The first is to create a 2x2 matrix according to eq. 5.15 for all possible thresholds \( \epsilon \). Subsequently, each confusion matrix is assessed by a certain measure derived from text categorization, e.g. \( \text{acc}(\epsilon) = \frac{a(\epsilon) + d(\epsilon)}{N_1 + N_2} \). Still, one has to get rid of this \( \epsilon \)-dependency to generate a solid effect-size measure of the feature. Therefore, numerical simulations have been carried out employing data sampled from various distributions, different data amounts and the overlap of the given distributions. The main focus of this analysis is how changes in the overlap of two distributions (change in effect-size) manifest in the \( \epsilon \)-dependent measure, e.g. \( \text{acc}(\epsilon) \). The final definition of the \( \epsilon \)-independent DESI will comprise the last part of this subsection.

**Metrics in text categorization** Feature selection is a popular method when it comes to text categorization, predominantly spam filtering. “In text classification, one typically uses a ‘bag of words’ model: each position in the input feature vector corresponds to a given word or phrase. For example the occurrence of the word 'free' may be a useful feature in discriminating spam email. The number of potential words exceeds the number of training documents by more than an order of magnitude. Feature selection is necessary to make large problems computationally efficient – conserving computation, storage and network resources for the training phase and for every future use of the classifier. Further, well-chosen features can improve classification accuracy substantially, or equivalently, reduce the amount of training data needed to obtain a desired level of performance. Opinions differ as why this is so, but it is frequently acknowledged.” [196]

In [196–199] several metrics for feature ranking in the text classification domain are compared\(^9\). The elements of the confusion matrix in eq. 5.15 correspond to the true positives (\( =a \)), i.e. the number of positive cases (e.g. spam) containing a certain word (e.g. 'free'), the false positives (\( =b \)), i.e. the number of negative cases containing a certain word, as well as false negatives (\( =c \)) and true negatives (\( =d \)). With regard to [196–199] the bi-normal separation, the odds-ratio and the information gain show high performances in a binary classification scheme. Furthermore, three other measures are analyzed as baseline-measurements: F1Power, accuracy and the risk-ratio. This is intended to validate or dismiss the superiority of the aforementioned metrics.

\(^9\) Yang and Pedersen [197] compared \( \chi^2 \)-statistics, information gain and mutual information and show slight superiority of information gain for text categorization
Table 5.2 shows the six metrics\(^{10}\) according to [196].\(^{11}\)

<table>
<thead>
<tr>
<th>measure</th>
<th>equation</th>
<th>measure</th>
<th>equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>OddsRatio</td>
<td>( \text{odd}(e) = \frac{a(e) \cdot d(e)}{b(e) \cdot c(e)} )</td>
<td>RiskRatio</td>
<td>( \text{risk}(e) = \frac{a(e) \cdot A_\epsilon(e) + 1 - A_\epsilon(e)}{A_\epsilon(e)} )</td>
</tr>
<tr>
<td>Accuracy</td>
<td>( \text{acc}(e) = \frac{a(e) + d(e)}{N_t + N_f} )</td>
<td>F1Power</td>
<td>( f_1(e) = \frac{2a(e)}{\text{acc}(e) + \text{risk}(e)} + c(e) )</td>
</tr>
<tr>
<td>Bi-normal separation</td>
<td>( \text{bin}(e) = \left</td>
<td>\text{cdfinv}(T(e)) - \text{cdfinv}(1 - S(e)) \right</td>
<td>)</td>
</tr>
</tbody>
</table>

Table 5.2: Overview of measures of performance based on the 2x2 confusion matrix

The information gain employs the mutual information content \( \mathcal{I}(x,y) \) given below:

\[
\mathcal{I}(x,y) = - \frac{x}{x+y} \log_2 \left( \frac{x}{x+y} \right) - \frac{y}{x+y} \log_2 \left( \frac{y}{x+y} \right) = \frac{1}{x+y} \left[ x \log_2 x + y \log_2 y - (x+y) \log_2 (x+y) \right]
\]

The Bi-normal separation makes use of the cumulative distribution function (CDF) for the normal distribution, more specifically, its inverse (see eq. 5.18; see eq. 26.2.23 in [200] for the exact constants). Whereas the CDF is a sigmoidal function identifying the abscissa with the cumulated density, its inverse is shaped like a tangent. It yields the abscissa for a given cumulated density. The major advantage of this non-linear transformation of PPV(\( e \)) and \( T(e) \), is that \( \text{cdfinv}(\text{PPV}(\epsilon)) \rightarrow \infty \) in case \( \text{PPV}(\epsilon) \rightarrow 1 \).

\[
x(p) = x|p = \text{CDF}(x) = t = \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3} \quad t = \sqrt{\ln \frac{1}{p^2}}
\]

Removing the \( \epsilon \)-dependency The generation of the final feature ranking score necessitates the removal of the \( \epsilon \)-dependency in the six functions of tab. 5.2. Hence, one searches for a score that assesses \( f(\epsilon) \). The confusion matrix and consequently the shape of each function \( f(\epsilon) \) is governed by the distribution of the samples from class one \( y_1 \) and the one from class two \( y_2 \). Clearly, \( f(\epsilon) \) will change depending on the shape of these distributions, the amount of data used and the amount of noise given. That is why several toy Monte-Carlo experiments have been carried out. They investigate whether there are characteristic changes in \( f(\epsilon) \) corresponding to changes in effect-size measures or not.

The samples \( y_1 \) and \( y_2 \) have been sampled from the same distribution, but with different parameterizations to allow different effect-sizes. In these experiments, four base distributions have been used: normal\(^{12}\), exponential\(^{13}\), logistic\(^{14}\) and uniform\(^{15}\) distribution.  

\(^{10}\) With respect to the odds-ratio, \( b(e) \) and \( a(e) \) have been increased by one in order to prevent zero-division (compare with original version in [196]). With regard to the risk-ratio, this holds only for \( c(e) \)  

\(^{11}\) For further mathematical justifications and derivations see [199]  

\(^{12}\) \( p(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \)  

\(^{13}\) \( p(x, \lambda) = \lambda \exp(-\lambda x) \)  

\(^{14}\) \( p(x, \mu, s) = \frac{e^{-(x-\mu)/\lambda}}{s^{1+e^{-(x-\mu)/\lambda}}} \)  

\(^{15}\) \( p(x, a, b) = \frac{a (e^{x-a} - e^{x-b})}{b - a} \)
5.2 Feature ranking by inference testing

Figure 5.2: bi-normal separation (a-c) and information gain (d-f) each calculated on five different random sample pairs from either a normal (a, d), an exponential (b, e) or an uniform (c, f) distribution. The plots show the respective functions together with the AUC of the dataset and the resulting DESI.
Chapter 5 Feature Selection

Accuracy, Information Power, F1 Power effect-size index The plots in fig. 5.2(d-f) show \( \inf(\epsilon') \) for two samples sampled from a normal(d), an exponential(e) and an uniform(f) distribution. As the shape of \( \inf(\epsilon') \) depends heavily on the underlying distribution, calculating the area under curve or even more advanced measures will be intractable. Therefore, the maximum of \( \inf(\epsilon') \) is used as the corresponding effect-size index. This also holds for the accuracy and F1 power effect-size index:

\[
INF = \max_{\epsilon' \in [1, \ldots, N_1+N_2]} \inf(\epsilon') \\
ACC = \max_{\epsilon' \in [1, \ldots, N_1+N_2]} acc(\epsilon') \\
F1 = \max_{\epsilon' \in [1, \ldots, N_1+N_2]} f1(\epsilon')
\]

Risk, Odds and binormal separation effect-size index One advantage of the latter indices is that they are naturally normalized between zero and one. This does not hold for the binormal separation \( \bin(\epsilon') \) depicted in fig. 5.2(a-c). Therefore, unnormalized maximum \( \text{BIN}_{\text{raw}} \), calculated analogue to eq. 5.19 is normalized according to:

\[
\text{BIN} = \frac{\text{BIN}_{\text{raw}} - \text{BIN}_{\text{rand}}}{\text{BIN}_{\text{tm}} - \text{BIN}_{\text{rand}}} 
\]

The normalization is based on the assumption that the lower bound of \( \text{BIN}_{\text{raw}} \) results from the contingency table being purely random, i.e. \( \begin{pmatrix} N_1 & N_2 \\ N_1 & N_2 \end{pmatrix} \). Plugging the latter into eq. 5.22 yields \( \text{BIN}_{\text{rand}} = 0 \). The upper bound of the bi-normal separation is determined by its value towards the optimal contingency table, i.e. \( \begin{pmatrix} N_1 & 0 \\ 0 & N_2 \end{pmatrix} \). One easily derives that \( \text{BIN}_{\text{tm}} = |\text{cdfinv}(1) - \text{cdfinv}(0)| \). The calculation of the odds and risk effect takes the same path. The only difference is that optimal and minimal values change: \( \text{ODD}_{\text{tm}} = (N_1 - 1) \cdot (N_2 - 1) \); \( \text{RSK}_{\text{tm}} = N_2 \); \( \text{ODD}_{\text{rand}} = \text{RSK}_{\text{rand}} = 1 \).

In summary:

\[
\begin{align*}
\text{BIN} &= \frac{\text{BIN}_{\text{raw}}}{|\text{cdfinv}(1) - \text{cdfinv}(0)|} \\
\text{ODD} &= \frac{\text{ODD}_{\text{raw}} - 1}{(N_1 - 1) \cdot (N_2 - 1) - 1} \\
\text{RSK} &= \frac{\text{RSK}_{\text{raw}} - 1}{N_2 - 1} \\
\end{align*}

\[
\begin{align*}
\text{BIN} &= \max_{\epsilon' \in [1, \ldots, N_1+N_2]} \bin(\epsilon') \\
\text{ODD} &= \max_{\epsilon' \in [1, \ldots, N_1+N_2]} odd(\epsilon') \\
\text{RSK} &= \max_{\epsilon' \in [1, \ldots, N_1+N_2]} \text{rsk}(\epsilon')
\end{align*}
\]

5.3 Feature subset search in the multi-class setup

The last section showed new powerful tools for judging the performance of a single feature in discriminating between one class and another one. Now, the main task is to find a combination of features,
5.3 Feature subset search in the multi-class setup

a feature set, that is capable of discriminating between several (C=13) classes. This section will be outlined as follows: First, I would like to stress some problems of feature subset search by means of feature ranking techniques. This part will be based on a more comprehensive investigation by Guyon and Elisseeff [178]. Afterwards, a standard procedure for multi-class feature selection, based on Fisher’s linear discriminant analysis will be explained. The section concludes with a supposedly optimal, yet computationally intense, method for finding a potent feature set developed in this thesis.

5.3.1 General remarks

Performing feature subset search raises the question of how features should be combined to generate high performance. Basically, one strives to reduce noise, diminish redundancy and increase complementary and decorrelation between the features within the subset. For this purpose one has to take into account that:

1. The curse of dimensionality should not be a plain excuse.  
2. Perfectly correlated features should be discarded.  
3. Feature complementarity and feature correlation are not contradictory.  
4. No feature is a priori useless.

The first statement refers to the fact that the curse of dimensionality is often stressed when it comes to the usage of hundreds of features. As Guyon and Elisseeff [178] puts it: “Noise reduction and consequently better class separation may be obtained by adding variables that are presumably redundant.” They show that features that are independent and identically distributed (i.i.d), even with moderate separation power, can perform well in combination. In consequence, a feature subset with a large amount of i.i.d features will reduce noise and transform the curse of dimensionality into “the blessing of dimensionality”. Unluckily, the second statement comes into play, i.e. redundant features which show strong correlation should be omitted. This begs the question: What is perfect or strong correlation? The major problem is that, according to statement three, features can be complementary but still have slight correlations. Statement four is much more radical about this: Let us take a closer look at the famous XOR - problem, depicted in fig. 5.3. Both features (x- and y-coordinate) show no separation power by themselves. In the two-dimensional plot, one can see a structure which can be easily classified. So, how can a combination of identical distributed features yield an easier classification task. The reason is class-dependent correlation. Obviously, the x- and y-coordinate are anti-correlated for class one and correlated in class two. If one would ignore the class-membership, the x- and y-coordinate were almost uncorrelated. Still, this is only true in case one uses the same amount of samples from class one and two. For the ECG-classification problem at hand, the normal beats will dominate class-independent correlation measures. Hence, a XOR-situation might be missed.

16 [201] the points of a high dimensional hypersphere are predominantly at the edges of the sphere
17 Refers to “Perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them.”[178]  
18 Refers to “Very high variable correlation (or anti-correlation) does not mean absence of variable complementarity.”[178]  
19 Refers to “a variable that is completely useless by itself can provide a significant performance improvement when taken with others.” and “Two variables that are useless by themselves can be useful together.”[178]
Chapter 5 Feature Selection

Figure 5.3: Visualization of the XOR-Problem. The right side shows the Box-Plots of identically distributed features (x- and y-coordinate) of two classes (black and red). The combination of the x- and y-coordinate on the left side shows a far more easy classification problem.

5.3.2 Approaches in the literature

How do common feature selection approaches account for the problems mentioned above. To answer this question, I want to show a representative example by Yu and Chen \[118\]. They use a two-staged setup that is pretty typical:

First, one has to calculate the correlation between two features as a measure of redundancy.\(^{20}\) If there is a high correlation, one has to decide which feature has to be eliminated from the feature set. For this decision, Fisher’s linear discriminant analysis (LDA) is employed.\(^{21}\) With regard to a C-class problem, it is defined as the relation between the inter- and intra-class scatter of the feature vectors \(f_i\) (see \[118, 185, 204\]):

\[
S_k = \frac{S_B}{S_W} \tag{5.26}
\]

\[
S_B = \sum_{k=1}^{C} N_k \left( \bar{f}'_k - \bar{f}'_i \right) \tag{5.27}
\]

\[
S_W = \sum_{k=1}^{C} \sum_{f_k \in \tilde{X}_k} N_k \left( f'_k - \bar{f}'_k \right) \tag{5.28}
\]

The feature reduction is as follows: First, calculate the \(F(F-1)/2\) correlations between two features according to Pearson:

\[
\rho_{kl} = \frac{\text{cov}(f'_k, f'_l)}{\sigma_k \sigma_l} \tag{5.29}
\]

\(^{20}\) Fleuret [202] developed a “fast binary feature selection method with conditional mutual information” which serves as a correlation measure and shows superior classification power over \[203\]. Current implementation only allows calculation on discrete values. “Extension to the case of continuous valued features and to regression (continuous valued class) is the center of interest of our current works.” Therefore, it is not used.\(^{21}\) Acir [61] use a measure similar to the fisher discriminaility, the divergence analysis, which in addition uses quadratic terms.

96
and then decide to remove the $k^{th}$ resp. $l^{th}$ feature with the lower Fisher discriminality from the feature set. Even though Yu and Chen [118] gained good results on feature reduction ($F_{\text{raw}} = 30, acc_{\text{raw}} = 97.53\% ; F_{\text{sel}} = 18, acc_{\text{sel}} = 97.28\%$), their strategy is flawed for several reasons:

First, the Pearson correlation coefficient is only intended to find linear correlations on normally distributed features. The correlations on a more general feature set will have non-linear correlations and features will have non-normal distributions. Moreover, the correlation coefficient $|\rho_{kl}|$ will be heavily dominated by the correlation in the class with the most patterns, namely the normal beat. There is the possibility that features are highly correlated for the majority of classes, but are clearly decorrelated for only a single class. The consequences of removing this very feature can be seen in [118].

Let us take a closer look at the results of Yu and Chen [118] and focus on the change in classifier performance if less features are selected by their approach. The reduction of features does not tremendously alter the sensitivity of the normal beats (from 97.83\% ($F = 30$) to 96.30\% ($F = 12$)), but the sensitivity of the VESCs suddenly drops from 95.00\% ($F = 18$) to 83.46\% ($F = 12$). What happened? In the author’s point of view, the following happened: Yu and Chen [118] use two features namely the ratio $RR_{-1}/RR_{-2}$ and $RR_0/RR_{-1}$ which are highly correlated for normal and LBBB beats because the RR-intervals do not change rapidly for non-VESC beats. But those features strongly decorrelate in the presence of a VPC. Even though the Fisher LDA guards against removal of one of those features, for $F = 12$ one or both are removed, resulting in a decline of VESC-sensitivity. The latter statements also hold for VPCs (drop: 96.72\% down to 93.79\%). The smaller decrease in VPCs sensitivity is due to the fact that the early depolarization caused by the VPC does not manifest itself in the RR-interval only. In addition, it alters the shape of the QRS-complex. Hence, it can be tracked via morphologic parameters.

The main problem when using correlation for feature reduction in a multi-class setup is that the implications of high correlations are quite ambivalent: The redundancy in a binary-classification task can be seen as a redundancy that should be removed for faster computation. But one could also see this redundancy as an accessory robustness feature. Moreover, even in binary-classification, inter-class correlation differences can be helpful for classification because they reveal that for certain values two features decorrelate, which leads to clear disjoint volumes in feature space. (see XOR-problem above)

That said, those appraisals also hold for more advanced correlation calculation methods like conditional mutual information (CMIM) [202] or symmetrical uncertainty (SU) [203]. Furthermore, CMIM and SU are intended for binary or discrete variables (e.g. image classification) and are hardly transferable to continuous variables. With regard to their performance compared to other selection methods like ReliefFS[205] and CorrFS[206], the SU approach shows superiority in only three of ten datasets by means of accuracy, even though computation time and feature set size are superior. Strategies like CorrFS are based on the core idea that: “Good feature subsets contain features highly correlated with the class, yet uncorrelated with each other.”[206] This approach is neither theoretically pleasing nor feasible practically. This thesis will show in numerical simulations that correlation with class is of minor feature ranking power. (see chapter seven) Furthermore, CorrFS would necessitate creating a 1.4 times 1.4 million sized matrix to hold the correlations for each between-feature correlation. Only calculating this matrix would take ages. Hence, selection methods governed by correlation-based indices are discarded.

5.3.3 A general approach

The pitfalls of prematurely identifying highly correlated with redundant features was not taken into account of any of the above strategies. They simply ignore those facts. Let us recall the XOR-problem from above. Two features that perfectly overlap can do a perfect separation of the classes in combination.
Moreover, they show class dependent correlation between each other. Finding such powerful feature sets is a great challenge especially in the multi-class and massive feature selection setting. Using the performance scores only will lead nowhere. A more sophisticated approach would be the following:

Let \( r_{ik} \) denote the rank of the \( i \)-th value of the first class of the \( k \)-th feature. Let \( \delta_{1ik} = 1 \) if the latter rank is smaller than \( N_1 \), literally speaking: "The \( i \)-th value is on the correct side". Let \( \delta_{1ik} = -1 \) if it is on the wrong side. The criteria for a good feature set of size \( F \) can be defined in the following way:

\[
\Omega(\{k_1, ..., k_F\}) = \sum_{i=1}^{N_1} \Theta\left( \sum_{k \in \{k_1, ..., k_F\}} -\delta_{1ik} \right) \tag{5.30}
\]

\( \Omega \) counts the number of values in class one that are more often on the wrong side than on the correct side taken all features in the subset into consideration. In a multi-class environment, one would have to extend \( \Omega \) to \( \Omega_{ab} \) denoting \( \Omega \) for the comparison between the \( a \)-th and the \( b \)-th class. The main goal in both situations, binary and multi-class, is to minimize \( \Omega \). In the example stated above, \( \Omega(\{1, 2\}) = 0 \) is optimal. Even though this approach seems powerful, the high amount of computation time and memory needed render this approach unfeasible. This approach has remarkable resemblances to a machine learning technique developed by Ataman et al. [207]. Their LP ranker seeks "to optimize the WMW statistic, we would like to have all positive points ranked higher than all negative points, ..." [207]. From this they develop a sophisticated approach to ranked support vectors, showing comparable results towards standard support vector machines. From this perspective, this thesis’ general approach might be an embedded feature selection method of a linear LP ranker. As demonstrated by other authors, the importance of a feature can be measured by the weights obtained from a linear support vector machine. Still, calculating those weights for a rank based SVM for 1.4 million feature is out of reach.

### 5.4 Novel methods on multi-class separation

In the last section, a variety of challenges and possible strategies towards multi-class feature filtering have been discussed. This section is devoted to novel approaches developed in this thesis. These methods have different degrees of sophistication. Still, they are all based on the interclass scoring table (IST). In easy terms, the IST is a matrix assessing the power of a certain feature to differentiate between instances of the \( i \)-th and those of the \( j \)-th class by means of a certain effect-size index \( EFF \). This three-dimensional matrix \( (F \times C \times C = 1,400,000 \times 13 \times 13) \) is a powerful tool for new innovations in feature selection. Nearly every multi-class-filter-based feature selection strategy can be described in the IST-framework. Hence, it is well suited for comparing new and old approaches on a common basis.

The first part of this section is, therefore, concerned with the definition of the interclass scoring table (IST). The rest of the section addresses the question of how this large matrix (\( \approx 11 \) billion non-redundant entries) can be processed in order to yield a small and high-performing feature set (20-60 features). Two simple schemes for this task are introduced in the second subsection: the best table (TopMean)\(^{22}\) and best-per-class (TopPC) strategy. The TopMean strategy focuses on the table as a whole, whereas the TopPC strategy measures the features capabilities with respect to a single class of interest. The TopMean scheme operates on the table level and the TopPC analyzes the columns of the IST. It is, therefore, straightforward to propose a method that concentrates on the entries of the IST. This is exactly what the interclass table-filling strategy (TableFill) does. It is explained in detail in the third subsection. Those three methods have been studied in detail with respect to performance and stability.

\(^{22}\) A generalized version of the Multi-AUC approach by [208]
5.4 Novel methods on multi-class separation

The TopPC strategy produces the most stable and high-performing feature sets. (see chapter seven) That is why further information concerning correlations has been added to the algorithm. This improves the performance at only a small expense of stability. The final feature selection technique, the correlation reduced TopPC strategy (TopPCCorr) is explained in the fourth subsection. The IST and all four feature selection methods are summarized in the scheme in fig. 5.4.
Figure 5.4: Summary of the four feature selection methods developed in this thesis: TopMean (left bottom), TopPC (middle top), TopPCCorr (middle bottom), TableFill (right). All are based on the interclass scoring table (top left).
5.4 Novel methods on multi-class separation

5.4.1 The interclass scoring table

The interclass scoring table (IST) is the basis of the algorithms explained later on. Below, I recall the definition from the mathematical definition in chapter one eq. 2.10:

$$\Xi(EFF): \Xi_{kij} = EFF(f_i^k, f_j^k)$$

The scoring table employs an effect-size index $EFF \in \{RSK, ACC, \ldots, FSH\}$ to measure the separation power of the $k$th feature with regard to the $i$th and $j$th class. In essence, the scoring table contains $C(C-1)/2$ non-redundant binary feature selection tasks. The $\Xi$ matrix for the problem at hand is a $1,400,000 \times 13 \times 13$-matrix. Therefore, the following algorithms should account for finite computation power.

5.4.2 TopMean and TopPC search

Let us discuss some basic feature selection strategies concerning the interclass scoring table: Best table (TopMean) and best-per-class (TopPC) search.

**Best table** The best table or TopMean approach focuses on the average discriminant power of a single feature. The easiest way to do this is to sum up all elements $\Xi_{kij}$ of the $k$th feature. Clearly, this is an unweighted average. Therefore, the discriminant power of a single entry is weighted equally, whether it shows high, intermediate or low discriminant power or not. This discounts the fact that high scoring features increase between-class scatter, whereas low scoring ones introduce much noise to the system. Hence, low scores should be penalized and high scores rewarded. Accordingly, averaging could be done via a non-linear transformation, according to eq. 5.31.

$$BT_{non-linear}^k = \sum_{i=1}^{C} \sum_{j>i}^{C} \tan \left( \frac{\pi}{2} \left( \Xi_{kij} - \frac{1}{2} \right) \right)$$

(5.31)

This would be quite problematic. This thesis will investigate and compare different effect-size indices which all treat the same data differently. Hence, an increase in one effect-size does not necessarily increase another index by the same amount. That is why this non-linear transform should be adapted to each effect-size index. It gets apparent later on that the distribution from which the data is sampled plays a crucial role for the assessment of each effect-size index. Therefore, defining a non-linear transform a priori, suiting each effect-size index, is not feasible. Moreover, choosing an individual non-linear transform will cause the final performance to be an interference between the strength of an effect-size index and the chosen non-linear transform. As the main focus lies on the performance of those effect-size indices, no weighting will be imposed. In conclusion, the best table score $BT_k$ for the $k$th feature is defined to be:

$$BT_k = \sum_{i=1}^{C} \sum_{j>i}^{C} \Xi_{kij}$$

(5.32)
Chapter 5 Feature Selection

The final feature selection is performed by searching for the best \( N_{\text{top}} \) features with regard to \( BT_k \). In fact, \( BT_k \) is sorted in descending order. The first \( N_{\text{top}} \) features in the generated sorted list comprise the final feature set \( F_{\text{sel}} \).

**Best-per-class** Perhaps averaging over all table elements introduces excessively high noise and redundancy to the selection process. Moreover, it is possible to miss features which are high-performing with respect to certain classes. Consequently, the second strategy, the TopPC approach, is to use the features with respect to a class of interest. The best-per-class score \( PC_{ki} \) of the \( k^{\text{th}} \) feature for the \( i^{\text{th}} \) class \( PC_{ki} \) is calculated via:

\[
PC_{ki} = \frac{1}{C} \sum_{j=1}^{C} \Xi_{kij} \quad (5.33)
\]

The feature selection process is as follows: First, \( PC_{ki} \) is measured for all \( F \) features and \( C \) classes. Second, \( PC_{ki} \) is sorted with respect to the \( i^{\text{th}} \) class in descending order. Afterwards, the best feature for each class is chosen to be in the final feature set. This selection procedure is repeated \( N_{\text{top}} \) (top count) times. Accordingly, \( C \) features, one per class, are selected in each iteration. In conclusion, the overall amount of selected features \( F_{\text{sel}} \) equals \( N_{\text{top}} \cdot C \).

### 5.4.3 Interclass scoring table filling

The assumption of the best table search (TopMean) is that each element in the scoring table should be rated independently and that no order should be imposed on them. In contrast, the best-per-class search (TopPC) implies order by combining element ratings with regard to the same class. Both features share the same problem. As both use averaging techniques, high values in a single element of the scoring table are neglected. The interclass scoring table filling (TableFill) proposed in this subsection will specifically account for this scenario.

**Table sorting** The starting point of TableFill is again the IST (the matrix \( \Xi \)). The first step is to sort the matrix along the first axis according to eq. 5.34 using the sorting function \( \lambda_{ij}(k) \) defined in eq. 5.35.

\[
\Xi'_{kij} = \Xi_{\lambda_{ij}(k)ij} \quad (5.34)
\]

\[
\Xi_{\lambda_{ij}(1)ij} \geq \Xi_{\lambda_{ij}(2)ij} \geq \ldots \geq \Xi_{\lambda_{ij}(k)ij} \geq \ldots \geq \Xi_{\lambda_{ij}(F)ij} \quad (5.35)
\]

Keeping in mind that

\[
\Xi'_{ij1} = \max_k \Xi_{kij} \quad (5.36)
\]

\[
\Xi'_{ijF} = \min_k \Xi_{kij} \quad (5.37)
\]
one can easily define a top feature list for each comparison \((i, j)\):

\[
T_{ij}(\epsilon) = \left\{ \Xi'_{1ij}, \ldots, \Xi'_{kij} \right\} \quad (5.38)
\]

\[
\Xi'_{1ij} - \Xi'_{(k+1)ij} > \epsilon \quad (5.39)
\]

In eq. 5.38, \(\epsilon\) defines a threshold that shows how big the difference towards the best feature can be in order to still be part of the top list. In consequence, \(\epsilon\) determines the size of the vector \(T_{ij}\). The easiest way to choose the feature set from this point would be to select only the best features in each comparison according to 5.40.

\[
\bar{F}_{set} = \left\{ \lambda_{ij}(1) \forall i, j | i < j \right\} \quad (5.40)
\]

Obviously, this strategy could lead right away to a situation in which one feature is strong in one comparison but pretty weak in others. To overcome this drawback, the \(\epsilon\)-region introduces some fuzziness which will be exploited in the next step of TableFill; namely the feature-voting.

**Feature voting** After all top lists \(T_{ij}\) are created, each feature gets a vote for every comparison in which it is in the top list. Spoken mathematically:

\[
\omega_k(\epsilon) = \sum_i \sum_{j > i} \left\{ \begin{array}{ll}
1 & {k \in T_{ij}(\epsilon)} \\
0 & {\text{else}} 
\end{array} \right\} \quad (5.41)
\]

As one can easily derive from eq. 5.41, \(\epsilon\) directly determines the votes for each feature \(\omega_k\). The fuzziness can be understood by looking at the lower and upper boundaries of \(\epsilon\). Setting \(\epsilon = 0\) will yield non-zero votes only for the best feature in a certain comparison \((i, j)\). On the other hand, setting \(\epsilon = 1\) results in non-zero votes for all features.

\[
\omega_k(0) = \sum_i \sum_{j > i} \left\{ \begin{array}{ll}
1 & {k = \lambda_{ij}(1)} \\
0 & {\text{else}} 
\end{array} \right\} \quad (5.42)
\]

\[
\omega_k(1) = \sum_i \sum_{j > i} 1 = C \cdot (C - 1) \quad (5.43)
\]

According to 5.42 and 5.43, the fuzziness parameter \(\epsilon\) describes the tradeoff between powerful, but too overly focused parameters, and weaker, but broader, features. The size of \(\epsilon\) depends heavily on the problem and data at hand, and has to be chosen by experiment.

**Table filling** The final selection of features is carried out by the so called table filling. Consider a \(C \times C\) matrix containing empty sets for each entry. This table is filled with the best features for the corresponding comparisons \((i, j)\). This ensures that the final feature set accounts for each comparison
(i,j) by at least one feature within the $\epsilon$-region. For ease of writing, let us denote the votes at the beginning of the table filling with $\omega^0_k$. The $\epsilon$ is omitted here for notational ease.

The filling algorithm starts with a $C \times C$ table which is filled with ones on the upper triangular matrix:

$$\Omega^0 = \begin{bmatrix}
0 & 1 & 1 & \ldots & 1 \\
0 & 1 & & \ldots & 1 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & & \ddots & \ddots & 1 \\
0 & 0 & & & \\
\end{bmatrix}$$

(5.44)

The construction of the feature sets is as follows: First, one selects the feature with maximal vote for the final feature set. At the beginning, $\overline{F^{0}_{\text{set}}}$ = $\emptyset$. 

$$\overline{F^{1}_{\text{set}}} = \overline{F^{0}_{\text{set}}} \cup \{k^0_{\text{max}} : k^0_{\text{max}} = \arg \max_k \omega^0_k\}$$

(5.45)

The feature with maximal vote accounts for many comparisons. In the next step, all those comparisons should be disregarded. This process is reflected by setting all tuples $(i,j)$ to zero which $k^0_{\text{max}}$ accounts for.

$$\Omega^1_{ij} = 0 | k^0_{\text{max}} \in T_{ij}(\epsilon)$$

(5.46)

Furthermore, all features that are in the top groups $T_{ij}$ of $k^0_{\text{max}}$ will be devoted for each membership. (see the visualization in fig. 5.4)

$$\omega^1_k = \omega^0_k - \sum_{i,j=1 \atop i \neq j} 1 | k, k^0_{\text{max}} \in T_{ij}, \text{ else}$$

(5.47)

As the iteration progresses, more and more table entries are filled with zeros and more and more features will have only contributions to already zero-filled entries, which leads to their rejection. At the end of the iteration, there will be some table elements not filled because the majority of features together addressed only 90% of the comparisons. In this state of iteration, features which in particular are strong only for a designated comparison (i,j) will fill the left open table entries. Following the scheme above, one can select a maximum of $C(C - 1)/2$ features. This would imply that the top lists are distinct sets, i.e. no top list shares any feature with another one. This can be due to $\epsilon \to 0$ or very specific features.

In summary, TableFill takes features into account that are strong upon a variety of comparisons, as well as parameters that are only powerful discriminators on specific comparisons. These fill the empty spots of feature sets generated by the TopMean or TopPC strategy.
Correlation based best per class

In chapter 6, it will be demonstrated that the TopPC strategy is a stable and high-performing feature selection method. This is accomplished without exploiting any correlations. Obviously, choosing the best features per class with regard to over 1.4 million features will inevitably select features that are, in part, highly correlated. It is, therefore, advantageous to interlock the TopPC strategy with a correlation assessment.

The procedure is straightforward: First, one has to select a search subset $\tilde{S}_i$ from the overall set of features $\tilde{F}$. This set comprises the best $N_s << F$ features with respect to class $i$:

$$\tilde{S}_i = \{k_1, k_2, \ldots, k_{N_s}\} \mid PC_{k_i} > PC_{k_{i'}} \forall k \in \tilde{S}_i \land \forall k' \in \tilde{F} \setminus \tilde{S}_i$$  \hspace{1cm} (5.48)

$\tilde{F}_s$ denotes the final set of features with respect to class $i$ at the $s^{th}$ iteration step. The initial set $\tilde{F}_1^i$ contains the best feature with respect to the $i^{th}$ class:

$$\tilde{F}_1^i = \{\arg\max_k PC_{k_i}\}$$  \hspace{1cm} (5.49)

The subsequent correlation step computes the absolute mean Spearman rank correlation between features in the final feature set $\tilde{F}_i$ and those in the search set $\tilde{S}_i$:

$$r_{ki} = \frac{1}{|\tilde{F}_i|} \sum_{k' \in \tilde{F}_i} r_{spear}(f_k, f_{k'}) \forall k \in \tilde{S}_i$$  \hspace{1cm} (5.50)

The next feature is selected to have a mean correlation $r_{ki}'$ lower than the threshold $r_{max}$. Moreover, it should exhibit a high mean effect-size $PC_{ki}$. Therefore, the rule for adding a new feature $f$ to the final feature set $\tilde{F}_i$ is:

$$\tilde{F}_{i+1} = \tilde{F}_i \cup \{f\} \mid f = \arg\max_{k \in R_i} PC_{k_i}$$  \hspace{1cm} (5.51)

$$R_i = \bigcup \{k' \mid r_{k'j} < r_{max}\}$$  \hspace{1cm} (5.52)

This iteration goes on until $s = N_{top}$. The final feature set is simply $\tilde{F}_{i+1} = \bigcup \tilde{F}_i$. In summary: First, one selects the best $N_s$ features with respect to a certain class $i$. Afterwards, the correlation between the selected features and those in the search set is calculated. The next feature is chosen from the features set $R_i$, which contains those with a correlation smaller than the threshold $r_{max}$. The feature with the highest performance $PC_{ki}$ in this set is added to the final feature set. This procedure goes on until $N_{top}$ features are chosen with respect to all $C = 13$ classes.
Chapter 6

Core Classifier

The last sections illustrated the way the data is prepared (preprocessing), the features that can possibly be extracted from the filtered signal and strategies to select the best suited features for building an elaborate classifier. The classifier itself is the main focus of this chapter. The first part of it is, therefore, devoted to the question: What is the best classifier for the task at hand? Clearly, superiority is a notion depending on the scope chosen. This thesis concentrates on feature extraction and selection. Thus, many tests will be performed with the purpose to evaluate different strategies by means of classification performance. That is why, the classifier should be robust with respect to parameterization, i.e. slightly suboptimal parameter choices will not significantly affect the assessment of the different feature extraction and selection strategies. Moreover, it should retain high performance and computational feasibility. The first section will argue in favor of employing support vector machines (SVM) using the generalized multi-class approach by Allwein et al. [209].

An advantageous property of the SVM is that, given one uses the popular Gaussian-kernel, only two major hyper-parameters need to be optimized: a regularization parameter $C$ and the smoothness of the high dimensional feature space $\gamma$. Nevertheless, performing a full grid-search in the $C - \gamma$-plane can be computationally intense, as each point in this plane necessitates a cross-validated assessment of a single SVM. This gets even more complicated because the multi-class SVM scheme employed herein needs up to 80 binary SVMs. It gets even better. One has to perform the grid-search and the training of all 80 SVMs to finally assess the performance of a single feature selection strategy configuration. The generation of all results in chapter six towards feature extraction and selection necessitated the computation of around 200,000 single SVMs in total. Clearly, many evaluations have been carried out for previous testing purposes only. A typical grid-search approach probes around 400 different $C - \gamma$ configurations.[210] Solving over 80 million (400 times 200,000) convex optimization problems with 5000 to 25000 data points is beyond the limits of the computational capabilities that have been available. Therefore, a scheme for discarding many $C - \gamma$ points had to be determined. The fundamental hypothesis of this thesis is that the notion of optimal kernel bandwidth in histograms and optimal $\gamma$ intersect. Basic justifications are given in this chapter. Experimental substantiation is given in chapter seven. 

Outline As already said, this chapter will start with a basic review of possible ECG-classification strategies in literature and will reason in favor of support vector machines. The second part is devoted to the support vector machine itself. This section will demonstrate the basic principles of the SVM, i.e. finding a hyperplane in a high dimensional space using non-linear mapping functions (kernels). I will sketch the derivations of Smola et al. [211], who showed that those mapping functions are in fact Green’s functions of a regularization operator. The Gauss kernel parameterized by $\gamma$ can be regarded as a low-pass filter corresponding to a regularization operator smoothing the high-dimensional feature space. Regularization is a key concept of the SVM, which goes back to works by [212].

1 see Rifkin and Klautau [213]
for smoothing histograms has been extensively studied by Shimazaki and Shinomoto [214]. The third section will show how this can be transferred to smoothing high-dimensional feature spaces and how this approach relates to other schemes of reducing the $C-\gamma$-plane. In essence, this thesis provides the opportunity to split the two-dimensional grid optimization into two simple one-dimensional hill climbing procedures. The first three sections focus primarily on binary support vector machines (SVM). The fourth section will shed light on a general approach by Allwein et al. [209] to combine those SVMs to form a multi-class SVM. The last section will conclude this chapter with presenting the final multi-class SVM setup that is used in this thesis.

6.1 Choosing a classifier

The easiest way to choose a classifier would be to search for a classifier that shows superiority in most of the datasets, freely available or proprietary. This approach is infeasible for two reasons: First, there is no comprehensive meta study comparing all classifiers on all datasets. Second, this procedure is in conflict with the lemma of the "No free lunch theorem"[185], which deliberately states: "There is no classifier that is superior on all datasets."[185] Again, one can relax the requirements of superiority by searching a classifier superior on comparable datasets, what this even might mean. This would require a supercomputer to crunch all the classifiers on many datasets and compare results. Still, comparability can only be granted if all the classifiers are well tuned, i.e. hyper-parameters have chosen in order for the classifier to work at its optimum. The NIPS challenge 2003 evaluated different feature selection methods for binary classification tasks. It "revealed that hyperparameter selection may have played an important role on winning the challenge. Indeed, several groups were using the same classifier (e.g. a SVM) and reported significantly different results. We have started laying the basis of a new benchmark on the theme of model selection and hyperparameter selection."[178]

6.1.1 Classifiers for ECG-classification

The classification of ECG-arrhythmias is still an ongoing question and many researchers search for new ways to accomplish this task by means of more complex or more elaborate classifiers. Therefore, trends in the machine-learning and pattern recognition community swiftly penetrate this area of research. Accordingly, this brief review will be split into the two main branches of machine learning: unsupervised and supervised machine learning. "Supervised learning methods develop rules for the classification of samples of unknown origin on the basis of a group of samples with known categories (known set). Unsupervised learning methods instead do not assume any known set, and the goal is to find clusters of objects, which may be assigned to classes."[215]

Unsupervised classification The major difference to supervised learning techniques like SVM or feedforward artificial neural networks (ANN) is that "in contrast to feedforward supervised ANN learning algorithms, which are trained to identify predefined features in the data, self-organizing networks(SOM) organize themselves according to the natural structure of the data - no external teacher is called for"[47]. So in essence, the clustering finds clusters of interest in the data and the physician has to match those with real world classes. The self organizing maps (SOM) show promising results in [47] and were also employed by [22, 30, 216]. Yet, SOM is a quite sophisticated clustering approach. Typical clustering applications like minimum distance classifiers (e.g. [120, 138]) simply define a metric (e.g. Euclidean metric, Mahalanobis distance) and calculate the distance between samples in the high-dimensional feature space. The final classification is performed by assigning a test sample to the closest cluster.
The major questions are: What kind of metric should be employed and what is the measure of proximity?

In noisy environments like the ECG, some researchers use fuzzy clustering (see [64, 66]), which yields probabilities of cluster membership rather than a strict assignment. This can be furthermore employed by a subsequent supervised classification step. Thus, those probabilities can be the inputs of a multi-layer perceptron (MLP), for example (see [117]). The usage of techniques such as learning vector quantization (LVQ) (see [30, 58]) and spectral clustering (see [141]) is low. In the early 90s and late 80s of the last century, template matching was quite on vogue [85, 217] and has been used particularly in the last years [84]. It is based on correlating a template waveform with the current waveform and opt for the most similar template. So, one could think of the correlation as a metric in clustering. Anyhow, template matching should be regarded as a supervised learning technique, because the templates contain class information.

**Supervised learning** Early attempts towards supervised learning were conducted using syntactic learning², e.g. Hidden Markov Models[222]. As they were rendered low performing, similar approaches have not been conducted in the last decade. In the late 80s and early nineties, computation resources were limited. Therefore, each classifier had to account for "two constraints ...: (i) the computation demand must be low; (ii) the classification accuracy should be moderately high. More often than not, these requirements are contradictory. A compromise is reached to select methods that are fairly sophisticated for accurate classification while not demanding excessive computation."

[85] This has not changed since. Even though computational power increased tremendously during the last decade, complexity and performance of elaborate classifiers rose simultaneously.

Accordingly, former approaches using simple scoring or discriminant functions [3, 63, 223] were rapidly superseded by more potent, but still linear approaches, e.g. linear discriminant analysis (LDA) [20, 20, 21, 66, 67, 83]³. Clearly, there are more advanced follow-ups like the quadratic discriminant analysis (QDA)²[20] or the generalized linear model (GLM).

Modern methods do not simply focus on the statistical properties of the underlying classes. They concentrate on the structure of the higher dimensional feature space. The k-nearest neighbor (kNN) rule, for instance, selects the k nearest neighbors⁵ and opts for the class that accounts for the majority of neighbors. This can be considered a local search algorithm, which is closely related to clustering techniques. In contrast to clustering, though, kNN exploits the class membership. Hence, it is considered a supervised approach. It was used by [65, 120] and [64, 66] showing non-superior performance compared to LDA and artificial neural networks (ANN). The latter is one of the most widely used classifiers at the moment. Even though de Chazal and Reilly [83] showed that for their setup LDA and ANN perform alike, ANN and its derivatives dominate the field.

Typically, one should immediately distance oneself from stating ANN and LDA perform alike. ANN are such a vast and diverse type of classifiers that one has to take optimization techniques, layer structure⁶, connection of the neurons, number of neurons and transfer functions⁷ at each node into consideration. In the early nineties, feed forward neural networks, which predominantly used back propagation⁸ were common [17, 35, 71, 83, 113, 128, 224–227]. In the last decade, more elaborate versions, like the multi layer perceptron (MLP) [36, 131, 228, 229], radial basis function neural networks (RBF-NN) [161] and fuzzy neural networks (FNN) [57, 91, 92, 111, 230], evolved.⁹ A detailed comparative review that relates different ANN structures to SOM and LVQ was written by Basu et al. [232]. Besides bare ANNs,

---

² see the review by [218] and [219–221] from [22] ³ see section 4.3 for LDA and its relation to the Fisher discriminant feature selection ⁴ LDA uses linear combinations of features, whereas GLM uses a neural network to generate a function that combines the features and a second function that weights those combinations (see [90]). ⁵ depending on the distance metric ⁶ see Hosseini et al. [116] for investigations about different structures of feed forward neural networks ⁷ sigmoidal functions [71] ⁸ Others [53, 122] use Levenberg-Marquardt optimization or a genetic algorithm [70] ⁹ Some rarely used types are complex valued ANN [231] and decision based neural networks[52]
many authors use hybrid systems, wherein an MLP is combined with fuzzy clustering[109, 166] or many MLPs form a committee of machines[31, 32, 55, 66, 114, 233]. In the last years, there have been efforts to utilize the structure of ANNs within FPGAs to improve computation speed and ease usage.[60, 177].

6.1.2 Support vector machine as the better option

The superiority of ANNs has been shown in a variety of setups: Bortolan et al. [233] used a committee of machines that outperformed LDA, which is supported by findings of Froese et al. [70] using a single ANN optimized by a genetic algorithm. In Bortolan et al. [66], performances of multiple ANNs showed superiority over kNN, LDA, and fuzzy clustering, in the case that enough samples are available. In the last subsection, I touched slightly upon the huge versatility one encounters when faced with ANNs. The set of hyper parameters is huge and every ANN subtype has its own niche. Maybe there is a more robust classifier, easily tunable, equally high-performing, but less easily subject to overparameterization. In the NIPS challenge 2003[178], a promising approach was predominant: Seven of nine top classifiers in the NIPS challenge used kernel methods, from which support vector machines (SVM) are the most popular. This finding is supported by Forman [196] who "performed a brief pilot study using a variety of classifiers, including, Naive Bayes, C4.5, logistic regression and SVM with linear kernel 11. The results confirmed findings that SVM is an outstanding method (YanLiu1999, Joachims 1998, Dumais 1998), and so the remainder of our presentation uses it alone." This is supported by findings of Ribeiro et al. [141] showing superiority of SVMs over MLP, Radial basis function NN and Naive Bayes. A recent study confirmed SVMs high performance on the field of DNA methylation classification. [234]12 In a recent study by Galar et al. [15], untuned SVMs easily compete with kNN and decision-trees (C4.5). With regard to the results in the next chapter using optimized Gaussian kernels, SVMs outperform the latter classifiers easily on the datasets provided by Galar et al. [15].

As already stated in the beginning of this chapter, tuning of those methods is crucial, i.e. performing an hyper-parameter selection finding the optimal number of neural nodes, transfer function or cost parameters and kernel functions. Zadeh et al. [68] compared MLP to RBF, probabilisticNN and SVM, and demonstrated that in the case that the SVM is well set up, it shows superior performance.13 The superiority over MLP that is moreover shown in [59] and a hybrid neural network [236] needs some mathematical substantiations. A study by [237] concludes: "In classification mode the unbeatable is SVM, while in regression better generalization ability possesses MLP ... The MLP network implementing the global approximation strategy usually employs very small number of hidden neurons. On the other side the SVM is based on the local approximation strategy and uses large number of hidden units." Furthermore, they state that due to its formulation as a quadratic optimization problem, "SVM algorithm is usually much quicker" for large datasets.14

All those findings conclude that SVM is a highly potent, easy to use and fast machine learning strategy. Moreover, it evinces strong multi-class capabilities which have been proven recently in a large comparative study carried out by Galar et al. [15].

10 The above list is surely not comprehensive as new methods like rough sets (see [99]) evolve rapidly and can not be covered in here. 11 each using the WEKA open-source implementation with default parameters 12 Muñoz and Gonzalez [235] used for representation of high-dimensional functional data 13 [24, 137] showed Rule-based decision tree clustering decision tree using J48-algorithm is potentially superior to SVMs and RBF networks in multi-class setups 14 The superiority over MLP is also demonstrated by Übeyli [130], but with less data
6.2 The support vector machine

In the last section, I argued in favor of using support vector machines as the core classifier. This decision is based on experiments carried out by many researchers in the field of ECG-classification. Now, this section is going to explain some basic principles of supervised classification techniques. In essence, one trains a classification rule on a limited number of training samples exploiting their class information. Afterwards, the performance is tested on a disjoint test-set. In the second part of the chapter, the rule generation process of the support vector machine is explained in detail. The basic concept is to find an optimal hyperplane in a high-dimensional feature space. This is done by maximizing the margin between the samples of two classes. This maximization problem can be expressed as a quadratic optimization problem, which is shown in the second subsection. As data is imperfect, it is not always feasible to find an optimal hyperplane separating all samples relating to both classes. Therefore, a regularization term tuned by the cost parameter $C$ is introduced. Literally, $C$ determines how strong the amount of misclassification should govern the minimization process. Hence, increasing $C$ will reduce training errors, but potentially reduce margin, thus allowing higher test set misclassification. This technique is called soft margin support vector machine and is the topic of the third subsection.

In the fourth subsection, I follow derivations by Rifkin and Klautau [213], showing that the SVM is a variety of the Tikhonov regularization[212]. This allows a deeper understanding of the connections between multi-layer perceptrons, least square support vector machines and soft margin support vector machines. In conclusion, those techniques share the same math, but employ different measures of error. In typical applications of support vector machines, the SVM searches for the hyperplane in a hyperspace that is sometimes of infinite dimensionality. This is achieved by transforming the data using kernels, satisfying Mercer’s theorem. The findings of [211] show that those kernels can be understood from a regularization perspective: as Green’s functions of regularization operators (see subsection five). From this point of view, kernels can be regarded as low-pass filters, smoothing the hyperspace claimed by the training samples.

After some practical considerations in the sixth subsection, this insight will be exploited in the next section to create a very fast hyper-parameter selection scheme, which is a key finding of this thesis.

6.2.1 Mathematical preface

This section will include some more math, which is slightly overlapping with the definitions from chapter two. Hence, this section provides both clarification and definitions. First, let us state the problem of binary classification: Suppose one gathered experimental data using lots of sensors, calculations and so on. Each of the $N_{	ext{total}}$ measurements provides a sampling point $x_i$ representing the outcome of $F$ different sensors and calculations. Hence, the feature space under investigation is of dimension $F$ and $x_i \in \mathbb{R}^F$.

Now, assume that those samples split into two groups (binary classification). Some samples potentially originate from process one and some from process two. For the sake of the argument, let us claim that there is a necessity to judge automatically which sample was created by which process. Thus, one searches for an induction algorithm $I$ satisfying:

$$y_i = I(x_i) = \begin{cases} 
1 & \text{sample is instance of class (process) one} \\
-1 & \text{sample is instance of class (process) two}
\end{cases} \quad (6.1)$$

The induction algorithm, though, is typically not trivial. Therefore, supervised classification uses a sub-
sample of those \( N_{\text{total}} \) samples to train the induction algorithm. The training set comprises \( l \) tuples \( \tilde{X} = \{(x_1,y_1),\ldots,(x_l,y_l)\} \). The major pitfall in training the SVM is overfitting, i.e. achieving a high accuracy in the training set, whereas the test set yields low performance. With regard to SVMs, overfitting occurs mainly due to improper hyper-parameter selection. The major hyper-parameters in SVM are the regularization parameter \( C \) and the width of the Gaussian kernel \( \gamma \). During training, one tries to find the optimal tuple \( (C,\gamma) \), ensuring good generalization capabilities. The generalization capabilities are assessed using a common cross-validation scheme which should guard against overfitting. \( k \)-fold cross-validation splits the training samples into \( k \) disjoint folds \( \tilde{X}_i \) and tests the performance of the corresponding induction algorithm \( I_i \) on the \( i \)th fold, previously excluded from training. More mathematically these folds satisfy:

\[
\tilde{X} = \{(x_1,y_1),\ldots,(x_l,y_l)\} = \bigcup_{i=1}^{n} \tilde{X}_i \quad \text{and} \quad \tilde{X}_i \cap \tilde{X}_j = \emptyset \quad \forall i,j
\]  

(6.2)

The induction algorithm used to test the \( i \)th fold is denoted:

\[
I_i = I(\tilde{X} \setminus \tilde{X}_i)
\]

(6.3)

The hyper-parameter selection can now be guided by the cross validation accuracy:

\[
\text{ACC}_{CV} = \frac{1}{l} \sum_{i=1}^{k} \sum_{x \in \tilde{X}_i} \delta(I_i(x) - y_i)
\]

(6.4)

After an optimal combination \( (C,\gamma) \) has been determined, the SVM is retrained using all samples \( \tilde{X} \).

### 6.2.2 SVM: Optimal Hyperplane

This thesis will employ support vector machines using the popular LIBSVM library[238]. More precisely the \( C \)-support vector classification (SVC) scheme is used. The following definitions of SVC are in accordance with those in the LIBSVM library [238], which is based on work by [239, 240]. The structure of the next two subsections is chosen according to chapter two in [240]. See fig. 6.1 for a visualization of the derivations to come.

The basic assumption of a support vector machine is that an optimal hyperplane exists which separates samples of class one from those of class two (blue resp. red in fig. 6.1). Mathematically spoken one searches for a hyperplane \( (w,b) \) under the constraint:

\[
y_i(w \cdot x_i + b) \geq 1
\]

(6.5)

This states that samples of class one \( (y=-1) \) should be above \( w \cdot x_i + b \leq -1 \) and those of class two \( (y=1) \) should be below the hyperplane \( w \cdot x_i + b \geq +1 \). Under these constraints, the optimal hyperplane is obviously given by:
6.2 The support vector machine

Figure 6.1: Schematic representation of the basic aspects of support vector machines.

\[ \text{Margin} = \frac{2}{\sqrt{w^T w}} \]

\[
\begin{align*}
&\mathbf{w}^T \mathbf{x} + b = -1 \\
&\mathbf{w}^T \mathbf{x} + b = 0 \\
&\mathbf{w}^T \mathbf{x} + b = +1 \\
&0 < \xi < 1 \\
&\xi > 1
\end{align*}
\]

Support Vectors (\(\xi = 0\))
The major task of the SVM training process is to determine \( w_0 \) and \( b_0 \). The performance of the SVM is evaluated by checking the hyperplane equation with new test patterns \( x'_j \), which should correspond to their true labels \( y'_j \):}

\[ y'_j = w_0 x'_j + b_0 \] (6.7)

The following paragraphs will show how the parameters \( w_0 \) and \( b_0 \) can be computed. In essence, the optimal hyperplane is comprised of the support vectors, i.e. all vectors for which eq. 6.5 turns into an equality. This means that the hyperplane is generated by the vectors of class one and two most proximate to those of class two resp. one.

**The primal optimization problem** How can one calculate \( w_0 \)? A nice property of the hyperplane \((w_0, b_0)\) is that it maximizes the distance between the two classes. Thus, the margin of the hyperplane \( \rho(w_0, b_0) \) is given by (see [239, 240]):

\[ \rho(w_0, b_0) = \frac{2}{|w_0|} \] (6.8)

In conclusion, finding \( w_0 \) is equivalent to maximizing the margin and consequently minimizing the norm of \( w_0 \):

\[ w_0 = \arg \min_w \frac{w \cdot w}{2} \] (6.9)

under the constraint eq. 6.5. This constrained minimization problem is also referred to as the primal problem.

**The dual problem: The Kuhn-Tucker equations** Finding a solution of the primal problem gets a lot easier with respect to its dual. In the framework of Lagrange multipliers, one can combine the minimization problem together with its constraints into a minimization problem of a single functional. Thus, the Lagrangian of the above minimization problem reads:

\[ L(w, b, \Lambda) = \frac{w \cdot w}{2} - \sum_{i=1}^l \alpha_i \left[ y_i (x_i \cdot w + b) - 1 \right] \] (6.10)

Referring to the Kuhn-Tucker theorem, the Lagrangian is optimal in the case that it is minimized with regard to \( w \) and \( b \), whereas it should be maximized with regard to \( \Lambda = (\alpha_1, \ldots, \alpha_l) \geq 0 \). (see [240])

\(^{15}\) Afterwards one can determine the confusion matrix and accuracy of the rule created.
6.2 The support vector machine

\[ \frac{\partial L(w, b, \Lambda)}{w} \bigg|_{w=w_0} = 0 \implies \sum_{i=1}^{l} \alpha_i y_i x_i = w_0 \] (6.11)

\[ \frac{\partial L(w, b, \Lambda)}{b} \bigg|_{b=b_0} = 0 \implies \sum_{i=1}^{l} \alpha_i y_i = 0 \] (6.12)

One can easily deduce from eq. 6.11 that the final \( w_0 \) is a linear combination of the training patterns \( x_i \) weighted by the Lagrange multipliers \( \alpha_i \). One easily infers that the support of the hyperplane is given by the fraction of training patterns with non-zero \( \alpha_i \). Hence, they are called support vectors. Now, let us finish the dual problem by plugging the Kuhn-Tucker equations eq. 6.11 and eq. 6.12 back into the Lagrangian. This yields the maximization problem:

\[ L(w_0, b_0, \Lambda) = \frac{w_0 \cdot w_0}{2} - \sum_{i=1}^{l} \alpha_i y_i x_i w_0 - \sum_{i=1}^{l} b_0 \alpha_i y_i + \sum_{i=1}^{l} \alpha_i \] (6.13)

\[ = \sum_{i=1}^{l} \alpha_i - \frac{w_0 \cdot w_0}{2} \] (6.14)

\[ = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \] (6.15)

\[ = \Lambda^T I - \frac{1}{2} \Lambda^T D \Lambda \] (6.16)

constrained by:

\[ \Lambda^T Y = 0 ; \; \Lambda \geq 0 \] (6.17)

using the distance matrix \( D \) containing the dot products of all pairs of training patterns:

\[ D_{ij} = y_i y_j x_i \cdot x_j \] (6.18)

The Kuhn-Tucker theorem, moreover, states that at the saddle point \( L(w_0, b_0, \Lambda_0) \) the equality below holds.

\[ \alpha_i^0 \left[ y_i (x_i \cdot w_0 + b_0) - 1 \right] = 0 \] (6.19)

Obviously, this equality holds for any \( \alpha_i^0 = 0 \). For any non-zero \( \alpha_i^0 \) eq. 6.5 turns into an equality. As \( w_0 = \sum \alpha_i^0 y_i x_i \), the hyperplane is generated by the training patterns with non-zero Lagrange multipliers. These support vectors are the training patterns closest to the optimal hyperplane. The dual problem sketched on the last pages can be easily solved by proper optimization strategies for quadratic problem...
6.2.3 Soft margin hyperplanes

The critical reader may have recognized that the above derivations assume that an optimal hyperplane exists. Clearly in most scenarios, perfect separation is impossible. Hence, one should allow for small errors $\xi_i$ for each training pattern $x_i$. Now the previous constraint eq. 6.5 is altered using the latter errors:

$$ y_i(w \cdot x_i + b) \geq 1 - \xi_i $$ (6.20)

This adds an error-term to the primal problem eq. 6.9:

$$ \min_w \frac{w \cdot w}{2} + C \sum_{i=1}^{l} \xi_i \quad \text{subject to } \xi_i \geq 0 \forall i $$ (6.21)

Equation 6.21 introduces the regularization parameter $C > 0$. It governs the trade-off between training error and generalization capabilities of the classifier. For $C \to 0$ eq. 6.21, the optimization tries to find the optimal margin excepting a large number of training errors. The classifier will generalize better with the consequence of lower performance. Increasing $C$ will enhance the importance of training errors. Taken to the extreme, the optimization will opt for a hyperplane which has practically no margin at all ($\frac{w \cdot w}{2} \to \infty \to 0$). Even though this can potentially decrease the training error. Nonetheless, the performance of this classifier will eventually drop off when facing an independent test set. Hence, generalization power of the classifier diminishes.

Again, this quadratic programming problem can be transferred to its dual convex optimization problem. Let us take a look at the Lagrangian of the constrained eq. 6.21:

$$ L(w, b, \Xi, \Lambda, R) = \frac{w \cdot w}{2} + C \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} a_i [y_i(x_i \cdot w + b) - 1 + \xi_i] - \sum_{i=1}^{l} r_i \xi_i $$ (6.22)

$$ = \frac{w \cdot w}{2} - \sum_{i=1}^{l} a_i [y_i(x_i \cdot w + b) - 1] + \sum_{i=1}^{l} \xi_i (C - a_i - r_i) $$ (6.23)

Now, one does not only have to optimize the multipliers $\Lambda$ concerned with the inequality 6.20. The multipliers $\Lambda$ ensure the positiveness of $\xi_i > 0$. Fortunately, the error term in eq. 6.21 comes in handy as all contributions of $r_i$ will cancel out. This can be derived by calculating the saddle point of the optimization of $(w, b, \Xi)$:
The support vector machine

\[ \frac{\partial L}{\partial w} \bigg|_{w=w_0} = w - \sum_{i=1}^{l} a_i y_i y_i = 0 \]  
(6.24)

\[ \frac{\partial L}{\partial b} \bigg|_{b=b_0} = \sum_{i=1}^{l} a_i y_i = 0 \]  
(6.25)

\[ \frac{\partial L}{\partial \xi_i} \bigg|_{\xi_i=\xi_i^0} = C - a_i - r_i = 0 \]  
(6.26)

The first equations are analogue to the optimal hyperplane case. The latter equation states that the sum of the "regularization" multiplier \( r_i \) and the "optimization" multiplier is constant \( C > 0 \). Accordingly the Lagrangian is:

\[ L(w_0, b_0, \Xi, \Lambda, R) = \Lambda^T \mathbb{I} - \frac{1}{2} \Lambda^T DA + \sum_{i=1}^{l} \xi_i (C - a_i - r_i) \]  
(6.27)

\[ = \Lambda^T \mathbb{I} - \frac{1}{2} \Lambda^T DA \]  
(6.28)

As the regularization and optimization multipliers are strictly positive: \( R \geq 0 \) resp \( \Lambda \geq 0 \), the maximization of \( L(w_0, b_0, \Xi, \Lambda, R) \) is constrained by:

\[ 0 \leq \Lambda \leq C \mathbb{I} \]  
(6.29)

\[ \Lambda^T Y = 0 \]  
(6.30)

In consequence, the difference between optimal hyperplanes and soft margin ones is that the introduction of an upper bound for the Lagrange multipliers \( \Lambda \). The regularization constant \( C \) determines how strong the sum of errors is weighted during optimization.

### 6.2.4 Tikhonov regularization

The last two subsections introduced the basic concepts of support vector machines: The training examples are separated by a hyperplane in the feature space. This hyperplane is optimized to guarantee maximum margin \( \rho = \frac{2}{w_0 \cdot w_0} \) between samples of class one and two. Hence, the necessary optimization minimizes \( w_0 \cdot w_0 / 2 \). To take into account errors during training, the soft margin introduces an upper bound for the Lagrange multipliers used for convex programming. This subsection is intended to generalize the findings of the linear SVM to non-linear situations. Moreover, light is shed on the regularization, which was parenthetically introduced via the sum of errors \( \xi_i \).

Let us first recall the primal problem of soft margin linear support vector machines:
minimize: \( \frac{\mathbf{w} \mathbf{w}}{2} + \sum_{i=1}^{I} \xi_i \)  
subject to: \( y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \)  

The problem above belongs to a much wider class of problems. One should consider that the optimal hyperplane optimization problem is ill-posed for many real-life data, i.e. there is no unique solution to it. Tikhonov and Arsenin [212] pointed out that ill-posed problems can be solved via regularization (see [213])

\[
\minimize: f(x_i) \cdot f(x_i) + \lambda \sum_{i=1}^{I} V(f(x_i), y_i) \tag{6.33}
\]

"\( V(f(x,y)) \) represents the cost we incur when the algorithm sees \( x \), predicts \( f(x) \), and the actual value is \( y \). The parameter \( \lambda \) is the 'regularization parameter' which controls the tradeoff between the two conflicting goals of minimizing the training error and ensuring that \( f \) is smooth"[213] One easily infers that the Tikhonov regularization yields the soft margin linear SVM using: \( f(x) = w \mathbf{x} + b \) and \( \xi_i = \max(1 - y_i f(x_i), 0) \), which is also referred to as the "hinge loss"[213]. In the view of the derivations of Rifkin and Klautau [213], other approaches to SVMs coined: least square support vector machines [241] or proximal support vector machines [244, 245] (see [213]) have all the same foundation. They are equal to the C-SVM approach, but use the "square loss" \( V(f(x,y)) = (f(x) - y)^2 \) to assess the classification error. Hence, this approach should be called: "regularized least squares classification" (RLSC) [213]. For most datasets, their accuracy does not differ. Therefore, RLSC will not be discussed further.

Work carried out by [247] shows, that perceptrons trained using the cross-entropy criterion use the same mathematical framework as SVMs. The major difference is that they use \( V(f(x,y)) = \log(1 + \exp(-f(x))) \) as their cost function.

### 6.2.5 The "kernel-trick"

With respect to the last subsections, the hyperplane remained a linear one. It would be remarkable if the great success of SVMs could be accomplished via a simple linear hyperplane \( f(x_i) = w_0 x_i + b_0 \). In fact, in many cases linear support vector machines lack high accuracy. Hence, a mapping function \( \phi(x) \) is introduced, which maps \( x \in \mathbb{R}^F \) into a higher-dimensional space \( \mathcal{F} \). In this particularly infinite-dimensional space, an optimal hyperplane \( w_0 \phi(x) + b_0 = 0 \) is more likely to be found. Consider that as \( \phi(x) \in \mathcal{F}, w_0 \) is element of \( \mathcal{F} \) too. This is a major problem if one wants to solve the primal soft-margin optimization problem eq. 6.21. The calculation of \( ||w||^2 \) is infeasible in a space of infinite dimension.

Providentially, the dual optimization problem leads to an easy way out. Let us recall it here:

---

16 Note that the equation below differs from [213] by the position of \( \lambda \), as the author wants to emphasize that the error is tuned, not the margin. 17 employed by [61, 242, 243]
maximize: $\Lambda^T \mathbb{1} - \frac{1}{2} \Lambda^T D \Lambda$ \hfill (6.34)
subject to: $0 \leq \Lambda \leq C \mathbb{1}$ \hfill (6.35)
$\Lambda^T Y = 0$ \hfill (6.36)

The influence of the map $\phi(x)$ on the outcome of the optimization is restricted to the calculation of the distance matrix:

$$D_{ij} = y_i y_j x_i x_j \rightarrow y_i y_j \phi(x_i)\phi(x_j) \hfill (6.37)$$

A direct and computationally intense calculation of $\phi(x_i)\phi(x_j)$ can be omitted. The only requisite is that a dot product $k(x_i, x_j) = \phi(x_i)\phi(x_j)$ exists. For this to be true, $k(x_i, x_j)$ has to be an integral operator kernel which is:

- symmetric, thus $k(u, v) = k(v, u)$
- an element of $L^2$
- in accordance to Mercer’s theorem

The latter theorem states: for $k(u, v)$ to be a dot product, it has to satisfy:

$$\int \int k(u, v)g(u)g(v)dudv > 0 \hfill (6.38)$$

for any square-integrable function $g(u) \in L^2$. The first two requisites, symmetry and square-integrability, allow us to expand $k(u, v)$ within the eigensystem $(\lambda_i, \phi_i)$ according to the Hilbert-Schmidt theory [248].

$$k(u, v) = \sum_{i=1}^{\infty} \lambda_i \phi_i(u) \cdot \phi_i(v) \hfill (6.39)$$

The eigensystem of the integral operator defined by the kernel $k(u, v)$ is generated via:

$$\lambda_i \phi_i(v) = \int k(u, v)\phi_i(u)du \hfill (6.40)$$

The Mercer theorem simply demands that all eigenvalues are positive ($\lambda_i > 0$), which is a necessary condition for $k(u, v)$ to define a dot product in $L^2$. In the end, one can employ any kernel $k(u, v)$ satisfying the above criteria and calculate the distance matrix $D$ without using the actual map $\phi(u)$:

---

\[ \text{The following explanations are according to section 4 in [240]} \]
\( D = y_i y_j \phi(x_i) \cdot \phi(x_j) = y_i y_j k(x_i, x_j) \)  

(6.41)

This allows us to solve the dual problem quite fast. However, new samples have to be evaluated without actually performing the map \( \phi(x) \). The kernel-trick also eases this part. Recall that the decision function is defined as \( f(x) = w \phi(x) + b \). Moreover, the hyperplane’s normal vector \( w \) is a superposition of the support vectors \( (\alpha_i > 0): w = \sum \alpha_i \phi(x_i) \). Plugging both equations together yields:

\[
f(x) = w \phi(x) + b = \sum_{i=1}^{l} \alpha_i k(x_i, x) + b = \sum_{i=1}^{l} \alpha_i k(x_i, x) + b
\]

(6.42)

### 6.2.6 The Gaussian kernel as a "low-pass filter"

Even though the “kernel-trick” eases calculation one still has no glimpse of the choice of \( k(u, v) \). Smola et al. [211] elucidated this notion by comparing support vector machines to regularization networks. In fact, both techniques try to minimize the empirical risk estimate \( V(f(x), y) \) (see eq. 6.33). The major difference between them resides in their regularization approach, towards the hyperplane \( f(x) = w \phi(x) + b \). Regularization networks try to optimize \( ||\hat{P}f||^2 \), where \( \hat{P} \) is a regularization operator. Support vector machines, on the other hand, minimize \( ||w||^2 \). Smola et al. [211] showed that these two approaches coincide, in case:

\[
k(u, v) = \phi(u) \phi(v) = (\hat{P}k)(u)(\hat{P}k)(v)
\]

(6.43)

Equation 6.43 is only satisfied if the kernel is the Green’s function of the operator \( \hat{P}^* \hat{P} \) [211]:

\[
(\hat{P}^* \hat{P}) k_u = \delta_u(x) \iff \phi: u \mapsto (\hat{P}k)(u)
\]

(6.44)

Still, this does not seem like an easy cookbook for new kernels. Nevertheless, it leads the way to a recipe. Recall that, as a symmetric function, one can represent the kernel \( k(u, v) \) in an eigensystem. Thus, one can choose this eigensystem to be a “subspace of the eigensystem of \( \hat{P}^* \hat{P} \)” [211], denoted \((\Lambda_i, \Psi_i)\). (compare with eq. 6.39)

\[
k(x_i, x_j) = \sum_i \frac{d_i}{\Lambda_i} \Psi_i(x_i) \Psi_i(x_j)
\]

(6.45)

In conclusion, any choice of \( d_i \in \{0, 1\} \) yields a new kernel corresponding to the regularization operator, as long \( \sum_i (d_i / \Lambda_i) \) converges.

**Example: Gaussian kernels** The latter formulation is quite general. To be more specific, let us take a look at a special class of kernels: translation invariant ones.

---

19 This subsection is a brief summary of the findings of Smola et al. [211]
Let us define a regularization operator which is a multiplication in Fourier space:\(^{20}\)

\[
\hat{P} \cdot \hat{f} = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\Omega} \frac{||\hat{f}(\omega)||^2}{P(\omega)}
\]  

(6.46)

One easily infers that the Fourier transform \(P(\omega)\) amplifies or dampens the Fourier transform of \(f(x)\): \(\hat{f}(\omega)\), which happens only within \(\Omega = \text{supp}[P(\omega)]\). The kernel can now be constructed solely based on the Fourier transform \(P(\omega)\):

\[
k(u, v) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} e^{i\omega(u-v)P(\omega)} d\omega
\]  

(6.47)

With regard to the Gaussian kernel:

\[
k(u, v) = \exp \left( -\gamma ||u - v||^2 \right) \bigg|_{\gamma = 2\sigma^2}
\]  

(6.48)

the Fourier transform \(\hat{P}(\omega)\) of the underlying regularization operator is \([211]\):

\[
P(\omega) = \exp \left( -\frac{||\omega||^2}{\gamma} \right)
\]  

(6.49)

This helps to understand how the Gaussian kernel works. It attenuates fluctuations in the high-dimensional space and therefore ensures smoothness of the high-dimensional feature space. This can also be understood from the perspective of gradient filtering: According to Smola et al. \([211]\) (see \([249]\) and Appendix A in \([250]\)), the corresponding regularization operator \(\hat{P}\) is\(^{21}\):

\[
||\hat{P}f||^2 = \int dx \sum_m \frac{1}{m! \gamma^m} \left[ (\Delta^m + \Delta \nabla \Delta^m) f(x) \right]^2
\]  

(6.50)

, where \(\Delta\) is the Laplacian and \(\nabla\) the gradient operator. From this perspective, the Gaussian kernel penalizes higher derivatives of \(f(x)\), again ensuring smoothness. The regularization capabilities of the Gaussian kernel are the key factor for boosting the \((C, \gamma)\) hyper-parameter search, which will be explained in section 5.3.

### 6.2.7 Practical considerations

So far, this subsection has focused on the theoretical foundations of the support vector machine. Still, one open question remains: How can the dual problem be solved in a real data scenario? Lots of

\(^{20}\) A more general formulation is used in the original paper by Smola et al. \([211]\). The formulation here is used for ease of understanding.  
\(^{21}\) The original paper\([211]\) is formulated using \(\sigma^2\). For notational convenience it is rewritten using \(\gamma = 2/\sigma^2\)
software libraries are evolving these days, as SVM gain more and more momentum with regard to their scope of application. The most prominent ones are LIBSVM [238], SVMlight [251, 252] and of course the implementation within the WEKA-library. I decided early-on in favor of LIBSVM, as it is widely employed[23, 33] and provides state-of-the-art and high-performing SVM calculation capabilities.

The details of fast SVM calculation using so called sequential minimal optimization (SMO)[253] are spared here. The interested reader is referred to [238, 251, 252] and literature mentioned there. Even though the software libraries take care of most of the work, the practitioner has to reflect on three major selection tasks:

- kernel selection
- normalization selection
- hyper-parameter selection ($C, \gamma$)

A general review focusing on the application of SVMs is provided by [210, 254]. Luts et al. [254] also states recommendations for weights in unbalanced data.

Kernel selection  Choosing the right kernel for the application at hand is crucial. According to section 5.2.6, the kernel $k(u, u)$ determines the smoothing / regularization properties of the non-linear map $\phi(u)$ that shapes the decision boundary of the SVM. Referring to [210], four kernels are quite common:

- linear kernel $k(u, v) = u \cdot v + c$
- polynomial kernel $k(u, v) = (\alpha u \cdot v + c)^d$
- Gaussian as an radial basis function kernel (RBF-kernel) $k(u, v) = \exp \left( -\frac{|u-v|^2}{2\sigma^2} \right) = \exp \left( -\gamma |u - v|^2 \right)$
- sigmoid kernel $k(u, v) = \tanh(\alpha u \cdot v + c)$

The Gaussian kernel or radial basis function kernel (RBF-kernel) is predominant in most ECG-classification applications [23, 34, 61, 108, 130]. Ribeiro et al. [141] substantiate this dominance with a comparative study in which they show that RBF kernels are superior to polynomial, linear and sigmoid kernels. This is also backed by findings of [124], who come to similar conclusions, but based on a lower amount of data. This is supported by Hsu et al. [210] stating, that the RBF-kernel has less hyper-parameters (only one: $\gamma$) than polynomial and sigmoid kernels. Moreover, linear and sigmoid kernels “behave like RBF for certain parameters"[255]”[210]. The usage of the RBF kernel eases computational stability and should only be omitted in the case that the amount of features greatly exceeds the number of instances. Even though, some authors use sigmoid [33] or wavelet kernels [256], this thesis will stick to RBF-kernels, as they show good performance in a wide range of applications.

Normalization selection  Still, there is a major obstacle for simply applying the Gaussian kernel to the raw data. Let us reflect on the distance metric used by this kernel:

$$||u - v||^2 = \sum_i (u_i - v_i)^2$$

22 see comparative study by Ribeiro et al. [141] 23 A more sophisticated approach using fuzzy support vector machines can be found in [108]
Clearly, the high-dimensional distance can be dominated by features which have a much wider numeric range or exhibit values at a larger scale than the others. This should be omitted in order to assure computational stability. Hence, Hsu et al. [210] recommends using a MinMax-normalization ([0, 1] or [−1, +1]) of the features. The major leap though, is that the normalization coefficients (minimum and maximum of each feature) have to be calculated during training. Therefore, the normalization procedure of the test data possibly yields values beyond [−1, +1] or [0, 1]. In chapter seven, a comparative study is carried out which investigates the differences between eleven normalization strategies:

- Max: \((x/x_{\text{max}})\), Min: \((x/x_{\text{min}})\), MinMax: \(((x - x_{\text{min}})/(x_{\text{max}} - x_{\text{min}}))\) norm
- Gauss: \(((x - x_{\text{mean}})/x_{\text{std}})\), Std: \((x/x_{\text{std}})\), TSig: \((\tanh(x - x_{\text{mean}}))\), GSig: \((\tanh(\text{Gauss}(x)))\) norm
- Log: \((\text{sgn}(x - x_{\text{median}}) \log(|x - x_{\text{median}}|))\), LSig: \((\tanh(\text{Log}(x)))\) norm
- Raw: \((x)\)
- Ranked norm

Most of the normalization techniques mentioned above are straightforward with the exception of the last one. The ranked norm takes into account that features contain much noise. Hence, it could be advantageous to restrict the possible values of a certain feature to a discrete set of numbers. Thus, all training samples of a given feature are ranked. The normalization of the highest feature value yields 1, the lowest yields 0 and all other \(r/N_{\text{train}}\), where \(r\) is the rank of the sample within the training set. During testing, one searches the closest training sample for a given test sample and identifies the test sample with the rank of the training sample. In essence, a training set consisting of \(N_{\text{train}}\) samples will yield a discrete set of \(N_{\text{train}}\) possible feature values.

Hyper-parameter selection  The hyper-parameter selection strategy will be explained in more detail in the next section, as it involves a major finding of this thesis.

6.3 Hyper-parameter-search via optimal histograms

Even though some authors [108] use fixed values for \(C\) and \(\gamma\), the typical approach is to carry out a grid-search for the optimal parameters [33, 124] like it is recommended by the LIBSVM maintainers [210]. This section will first explain the basic properties and methods of the grid search. It primarily measures the generalization error for several combinations of \(C\) and \(\gamma\) and employs the best configuration later on. As each tuple \((C, \gamma)\) enforces a recalculation of the SVM, this task is quite time consuming. In the setup of Hsu et al. [210], 441 SVM runs are necessary. In a multi-class classification of \(N\) classes, one needs \(N^2 + N(N-1)/2\) runs. Thus, in the worst-case scenario approx. 34,000 SVM’s have to be probed, given \(N = 13\). Obviously a strong reduction is necessary.

Two major reduction approaches have been proposed by [26] (54 SVM runs per grid search) and [27] (21 SVM runs). The proposal of Keerthi and Lin [26] shows that the two dimensional grid can be split into two successive one dimensional optimizations. They reveal a link between the kernel width \(\gamma\), the cost parameter \(C\) of the kernel-based SVM and the cost parameter \(\hat{C}\) of a linear SVM:

\[
\log(\gamma^{-1}) = \log(\hat{C}) - \log(C)
\] (6.52)
One can simply derive $\log(\tilde{C})$ by tuning the cost parameter of a linear SVM (no $\gamma$ optimization needed). Afterwards, one can tune $\gamma$ and adjust $C$ according to the above formula. The strategy of [27] is even more efficient. They give a confidence interval of $\gamma$ derived from the minimal distance between two samples in the training set. Furthermore, they exploit the properties of the shape of the generalization error in the $(\log C, \log(\gamma))$-plane. (see fig. 6.2)

Still, there is a more efficient way. This thesis will reveal that the confidence interval of $\gamma$ proposed by [27] brakes down to a single $\gamma$ value. $\gamma$ can be deduced by using a recent advance in histogram optimization strategies provided by [214, 257]. My hypothesis is that the optimal $\gamma$ corresponds to the optimal bandwidth $\sigma^2 = 1/(2\gamma)$ of a Gaussian kernel employed to provide a kernel based density estimation of the Euclidean distance between the training samples. Justification is provided by the success of the optimization scheme. It allows to optimize $C$ and $\gamma$ separately. Hence, the early stopping criterion by [26] is easily applicable.

These two approaches will be explained in the second part of this section. It concludes with this thesis’ proposal for a faster hyper-parameter search in the third subsection.

### 6.3.1 Grid search

The full-grid search proposed by Hsu et al. [210] basically searches for the optimal configuration $(C, \gamma)$ within a grid of $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$. The performance of each configuration is assessed via $n$-fold cross validation. Thus, the training set $D$ is split into $n$ disjoint subsets (see [258]):

$$D = \{(x_1, y_1), \ldots, (x_l, y_l)\} = \bigcup_{i=1}^{n} D^i \text{ and } D_i \cap D_j = \emptyset \forall i,j \quad (6.53)$$

The subset that includes the $i^{th}$ sample $(x_i, y_i)$ is denoted as $D^i(i)$. The $n$-fold cross validation estimate of the accuracy of a support vector machine is given by [258]:

$$\text{acc}_{CV} = \frac{1}{l} \sum_{(x,y) \in D} \delta(\text{SVM}(D\setminus D^i), x, y) \quad (6.54)$$

In practice, one trains $n$ SVMs each with the subset $D^i$ removed from the training set $D$. Afterwards, this SVM (SVM$(D \setminus D^i)$) is used to test all samples $x$ in $D_i$ whether they are in accordance with their respective labels $y$ or not. Thus, each sample is evaluated only once.

The extreme case of cross validation is the leave-one-out estimate, which is essentially setting $n = l$. This means $l$ SVMs are trained only removing one sample and testing this sample afterwards. This is a computationally intense procedure. Even though great speed ups can be accomplished using faster evaluation proposed by [251, 252], the cross validation is still faster. Moreover, for $n = 10$ or $n = 20$ the cross validation is quite unbiased [258]. Other approaches to error estimation, like boot-strapping show stronger bias, which is potentially too too optimistically biases.

24 Clearly, other estimates like recall, precision or F1score can be calculated this way. Nonetheless, they play a minor role in real-life applications.
6.3 Hyper-parameter-search via optimal histograms

The selection of the number of folds used for the assessment is crucial. Kohavi [258] recommends setting \( n = 10 \). Furthermore, one has to consider the stability of the induction algorithm (SVM) and the ratio of the sizes of the training and test split. Kearns [259] shows that the generalization error strongly depends on the amount of training data and the complexity of the algorithm. Even though they give some recommendations, one still has to determine experimentally which number of folds is optimal for the induction algorithm and the data at hand. With regard to the findings of [258], the bias of the cross-validation algorithm strongly depends on both data and induction algorithm.

The grid search proposed by [210] necessitates many runs of the cross-validation (SVM). Hence, Oommen et al. [261] recommend using the grid-search proposed by [26]. Hsu et al. [210] does not concur and states two reasons in favor of his extensive search: “One is that, psychologically, we may not feel safe to use methods which avoid doing an exhaustive parameter search by approximations and heuristics. The other reason is that the computational time required to find good parameters by grid-search is not much more than that by advanced methods since there are only two parameters. Furthermore, the grid-search can be easily parallelized because each \( (C, \gamma) \) is independent. Many of advanced methods are iterative processes, e.g. walking along a path, which can be hard to parallelize.” [210]

The last point is surely the most important. With the advances in modern multi-core CPUs, the difference between both grid searches can go from 441 vs. 54 SVM cross validations to 100 to 54 runs, given a quad-core CPU. Hence, the pure number of runs does not tell all about the computation time. Implementation and parallelization should be considered.

6.3.2 The fast grid approach

The fast grid approach by Keerthi and Lin [26] exploits the asymptotic behavior of SVMs employing the Gaussian kernel. Via real-data experiments one can verify that the generalization capabilities are restricted to a “good region” in the \((\log(\sigma^2), \log C)\) plane. This is due to four general asymptotic behaviors, which are proven by [26], depicted below. Besides the asymptotic behavior leading to underresp. over-fitting the most important feature in the plot is the straight line \( \log \sigma^2 = \log C - \log \tilde{C} \). This corresponds to the fact that, “if \( \sigma^2 \to \infty \) and \( C = \tilde{C} \sigma^2 \) where \( \tilde{C} \) is fixed, then the SVM classifier converges to the Linear SVM classifier with penalty parameter \( \tilde{C} \)” [26].

This asymptotic can be easily exploited using the approach of Keerthi and Lin [26]:

- perform grid search for \( \log \tilde{C} \in [ -8, 2 ] \) using a Linear SVM (yields \( \log \tilde{C}_{opt} \))
- perform grid search for \( \log \sigma^2 \in [ -8, 8 ] \) and \( \log C = \log \sigma^2 + \log \tilde{C}_{opt} \)

In consequence, the two-dimensional grid search is split into two one-dimensional ones. This accelerates calculation. In the setup in [26], they use a 0.5 resolution for each variable, which cumulates to 54 SVM runs. The only curtailment with regard to parallelization is that the first grid search can not be parallelized with the second one. If one relaxes the search to a resolution of one, this takes only 27 runs. This would also be a better comparison to the grid search proposed by [210].

\(^{25}\) For further information about necessary stability criteria of the induction algorithm see [260]
Figure 6.2: Scheme of the good region approach by Keerthi and Lin [26]. The picture is essentially like in [26], but with some extra clarifying remarks.
6.3 Hyper-parameter-search via optimal histograms

6.3.3 Intelligent grid determination

The latter grid search approaches focused on reducing the search area of the regular grid. Still a regular spaced grid is not the only possible choice. One should be aware that the “good region” depicted above forms a plateau in the \((\log C, \log \sigma^2)\)-plane. Moreover, the basic shape of the cross-validation accuracy is quite smooth. The noise level at the plateau is quite modest. Thus, a grid point can either be at the bottom of the “hill”, at the slope or the top. It is advantageous that the slope comprises only a small fraction of the \((\log C, \log \sigma^2)\)-plane. The majority of tuples \((\log C, \log \sigma^2)\) is either at the top or the bottom. Therefore, the grid can be severely thinned.

The strategy for thinning found by [27] is based on the pioneering work of [262], who developed a scheme for reducing measurement points. Their so-called uniformity designs (UD) resample the grid to gain a full coverage of the parameter-space with less grid points. More precisely, given a grid of size \(r \times r\), uniformity designs distribute \(r\) points over the search area. These points are chosen to maximize the \(L_2\)-discrepancy of non-uniformity. If one considers the “Hsu grid” \((\log C = [-10, 10] \rightarrow r = 21)\), a nice reduction from 441 to 21 grid points is achievable.

Since this grid could be too porous, Huang et al. [27] uses nested uniform designs. This means that first an even rougher grid is used \(r = 13\) for determination of optimal accuracy. Then the optimal grid point is chosen to be the center of the next uniform design (see pictures below) with \(r = 9\). The area covered is much smaller and, therefore, the search finer. As the center is not calculated twice, this yields \(r = 13 + 9 - 1 = 21\) SVM runs for searching the \((\log C, \log \sigma^2)\)-plane.

This amount of SVM runs can further be lowered by determining a lower and upper bound of \(\gamma\). This bound is data-dependent and is based on the following deliberation: Let us consider the pair of the closest distinct points \(x_i^*\) and \(x_j^*\) defined by:

\[
\rho^2 = ||x_i^* - x_j^*|| = \min_{x_i \neq x_j} ||x_i - x_j||^2
\]  

(6.55)

The Gaussian kernel measures the similarity of this pair calculating \(\exp(-\gamma \rho^2)\). So the question arises: how similar should distinct points be? Clearly, a high similarity could possibly blur all detail information causing under-fitting. A lower similarity, on the other hand, causes all points to be quite distinct from one another, leading to over-fitting. Thus, Huang et al. [27] restrict the similarity according to:

\[
0.15 \leq \exp(-\gamma \rho^2) \leq 0.999
\]  

(6.56)

This can be transferred to lower and upper bounds of \(\sigma\) and of \(\log_2(\sigma^2)\) (see [27])

\[
L_{\text{sigma}} \approx 0.5 \cdot \rho \quad \text{and} \quad U_{\sigma} \approx 22.4 \rho
\]  

(6.57)

\[
-2 \leq \log_2(\sigma^2) - \log_2(\rho^2) \leq 9
\]  

(6.58)

The latter equation shows that the search range of \(\log_2 \sigma^2\) shrinks, from \([-10, 10]\) to \([-2,9]\) (center can be shifted due to \(\rho^2\)). This reduced search space can be covered by smaller UDs. Hence, the final

\[\text{the second line of the equation was calculated within this thesis}\]
setup of [27] uses a 9 and 5 scheme similar to the one above.

### 6.3.4 Optimal histograms

The latter approach is a strong hint that optimal \( \sigma^2 \), and therefore \( \gamma \), corresponds to properties of the distribution of the distances between the samples within the data. The approach of [27] used lower and upper bounds derived from the minimal distance between training samples. Yet, this is still a quite rough estimate. In the next paragraphs, I will show a strategy to narrow \( \gamma \) down to a single, supposedly optimal value. In the subsection about regularization operators, it has been demonstrated that applying the kernel function corresponds to low-pass filtering/smoothing of the high-dimensional space. Its width in the frequency-domain is determined by \( \gamma \). In this sense, the bounds of \( \gamma \) can be understood as lower and upper bounds of the "stop" band of the low-pass filter.

**The concept of optimal time-histograms** The idea for the following estimation refers to the estimation of smoothed kernel densities for neural spike analysis. "In neurophysiological experiments, neuronal response is examined by repeatedly applying identical stimuli. The recorded spike trains are aligned at the onset of stimuli, and superimposed to form a raw density"[214]:

\[
\rho_t = \frac{1}{n} \sum_{i=1}^{N} \delta(t - t_i) \tag{6.59}
\]

"where \( n \) is the number of repeated trials."[214] Obviously, this density is pretty sparse and discontinuous. Hence, a kernel \( k(s) \) is applied to smooth this density, which yields the kernel density:

\[
\hat{\rho}_t = \int_{\mathbb{R}} \rho_{t-s} k(s) ds \tag{6.60}
\]

The kernel is defined to depend on the inter-point distance only and should satisfy:

\[
E(s^0) = \int s^0 k(s) ds = 1 \text{ and } E(s^1) = 0 \text{ and } E(s^2) < \infty \tag{6.61}
\]

In conclusion, the kernel \( k(s) \) should be normalized, symmetric and bounded. A good candidate is the familiar Gaussian kernel:

\[
k_{\gamma}(s) = \sqrt{\frac{\gamma}{\pi}} \exp \left( -\gamma \cdot s^2 \right) \tag{6.62}
\]

One easily infers that using larger \( \sigma \) will cause a very smooth estimate. As \( \sigma \) approaches zero, the kernel becomes the \( \delta \)-function. Thus, one searches for a trade-off between high resolution and smoothness of the kernel density. This optimization of the kernel has been carried out by [214] in a computationally cheap way, which allows one to estimate \( \sigma \) quite easily. Their approach is based on their work on optimal
6.3 Hyper-parameter-search via optimal histograms

histograms Shimazaki and Shinomoto [257][263], where the square function is used as the kernel. The histogram is considered optimal in case the mean integrated square error (MISE) is minimal:

\[
MISE = \int_a^b E(\hat{\rho}_t - E\rho_t)^2 dt
\]  

(6.63)

Shimazaki and Shinomoto [214] derive a general cost function, which should be reduced to gain minimal MISE.

\[
\hat{C}_n(\sigma) = \frac{1}{n^2} \sum_{i,j} \psi_\sigma(t_i, t_j) - \frac{2}{n^2} \sum_i \neq j k_\sigma(t_i - t_j)
\]

(6.64)

\[
\psi_\sigma(t_i, t_j) = \int_a^b k_\sigma(t - t_i) k_\sigma(t - t_j) dt
\]

(6.65)

With regard to the Gaussian kernel, this equation can be simplified using the symmetry of the kernel (\(t_i\) and \(t_j\) are exchangeable). The summation over \(i \neq j\) is simply a summing two times over \(i < j\), as the summands are equal for \(i > j\). The sum over \(\psi_\sigma\) can also be split into the sum over equal and unequal indices:

\[
n^2 \hat{C}_n(\sigma) = \sum_i \psi_\sigma(t_i, t_i) + \sum_{i < j} \psi_\sigma(t_i, t_j) - 4 \sum_i < j k_\sigma(t_i - t_j)
\]

(6.66)

Now \(\psi_\sigma(t_i, t_j)\) can be approximated by (see [214]):

\[
\psi_\sigma(t_i, t_j) \approx \frac{1}{\sqrt{\pi} 2\sigma} e^{-\frac{(t_i - t_j)^2}{4\sigma^2}}
\]

(6.67)

This leads to a tractable cost-function:

\[
2\sqrt{\pi n^2 \hat{C}_n(\sigma)} = \frac{N}{\sigma} + \frac{2}{\sigma} \sum_{i < j} \left\{ \exp\left(-\frac{(t_i - t_j)^2}{4\sigma^2}\right) - 2\sqrt{2} \exp\left(-\frac{(t_i - t_j)^2}{2\sigma^2}\right) \right\}
\]

(6.68)

**High dimensional space density** Now one has a good chance of finding a suitable \(\sigma\) to optimize a time histogram. What are the consequences of the problem at hand? Clearly one can define a raw space density similar to \(\rho_t\):

\[
\rho_x = \frac{1}{b} \sum_{i=1}^{N} \delta(x - x_i)
\]

(6.69)
Chapter 6 Core Classifier

Here the delta-function is of higher dimensionality, as \( x \in \mathbb{R}^F \). Again each sample point is “convolved” with a hyper-ellipsoid formed by the multi-dimensional Gaussian \( k_\sigma(Delta) = \exp(-\frac{|x|^2}{2\sigma^2}) \):

\[
\hat{\rho}(x) = \int_{\mathbb{R}^F} \rho x - \Delta k_\sigma(\Delta)
\] (6.70)

From this point all derivations are analogue to the findings of [214], which yields the cost function:

\[
n^2 \hat{C}_n(\sigma) = \sum_i \psi_\sigma(x_i, x_j) + 2 \sum_{i<j} \{ \psi_\sigma(x_i, x_j) - 2k_\sigma(x_i - x_j) \}
\] (6.71)

The major question is: does a tractable approximation of \( \psi_\sigma(x_i, x_j) \) exist. The properties of the Gaussian kernel allows one to easily extend it to higher dimensions. Let us take a closer look:

\[
\psi_\sigma(x_i, x_j) = \int_{\mathbb{R}^F} k_\sigma(x - x_i)k_\sigma(x - x_j)
\] (6.72)

\[
= \int dx_1dx_2 \ldots dx_F \exp\left(-\frac{1}{2\sigma^2} \left[ ||x - x_i||^2 + ||x - x_j||^2 \right]\right)
\] (6.73)

\[
= \int dx_1dx_2 \ldots dx_F \exp\left(-\frac{1}{2\sigma^2} \left[ \sum_{k=1}^F (x_k - x_{ik})^2 + (x_k - x_{jk})^2 \right]\right)
\] (6.74)

\[
= \prod_{k=1}^F \int_{\mathbb{R}} dx_k \exp\left(-\frac{1}{2\sigma^2} \left[ (x_k - x_{ik})^2 + (x_k - x_{jk})^2 \right]\right)
\] (6.75)

\[
e \approx \prod_{k=1}^F \exp\left(-\frac{(x_{ik} - x_{jk})^2}{4\sigma^2}\right)
\] (6.76)

\[
= \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)
\] (6.77)

**The algorithm** The major tool for finding an optimal \( \sigma^2 \) is the \( Cs \) function:

\[
Cs(\sigma^2) = \sum_{i=1}^{N_b} \sum_{j>i} e^{-\frac{|x_i - x_j|^2}{4\sigma^2}} - \sqrt[8]{8 e^{-\frac{|x_i - x_j|^2}{2\sigma^2}}}
\] (6.78)

- define a search range for \( \log_2(\sigma^2) \), e.g. [-20, 20] with resolution 1
- generate a boot strap estimate of \( Cs \) using \( b_{min} \) to \( b_{max} \) boot sets for each \( \sigma^2 \)
- set \( \log_2(\sigma_{opt}^2) = C_{smin} \)

130
6.4 Multi-class support vector machines

The latter subsection extensively described strategies to acquire an optimal binary SVM. Obviously, the problem at hand (ECG-classification) has more than two classes, namely 13. SVM multi-class classification can basically be divided into two branches: “all-together” methods and classifier ensembles [264]. The latter combine the results of many binary SVMs, whereas the first extends the binary SVM framework to multiple class labels. The task of selecting a fitting multi-class scheme is cumbersome. There is no theoretical explanation, or even proof, that any scheme will be superior for all or even the majority of datasets. Hence, one is left with experiments on real and simulated data. This thesis will omit discussing the “all-together” methods as they show comparable performance, but at the expense of higher complexity with regard to easier classifier ensemble techniques. (see [264])

6.4.1 The one vs. all (OVA) scheme

First, let us define the problem at hand: Consider a $C$-class problem with training samples \( \{(x_1,y_1), \ldots, (x_l,y_l)\} \). In the multi-class case $C > 2$, the labels are defined by $y \in \mathcal{Y} = \{1, 2, \ldots, C\}$. A simple approach to gaining a multi-class ensemble is the “one-vs-all” (OVA) approach. One trains $C$ different SVMs. The $k$th SVM is trained to separate the samples of the $k$th class from all other samples. Therefore, the labels $y_i$ are relabeled to apply to the $k$th binary SVM. The label $\hat{y}_k^i$ of the $i$th sample of the $k$th SVM can be obtained via:

\[
\hat{y}_k^i = \begin{cases} +1 & \text{if } y_i = k \\ -1 & \text{else} \end{cases} \quad (6.79)
\]

Thus, samples with $y_i = k$ are labeled $\hat{y}_i = +1$ and all others $\hat{y}_i = -1$ for the binary SVM. The result is that each class is labeled $+1$ only once and $C - 1$ times labeled as other, i.e. labeled as $-1$. During testing, a sample with $y_i = k$ should be labeled by the $k$th SVM with $+1$ and $-1$ for all others.

This can be described using a $C \times C$ matrix $M(k_1, k_2)$, which is shown below for the $C = 5$ case.

\[
M_{OVA}(k_1, k_2) = \begin{pmatrix}
+1 & -1 & -1 & -1 & -1 \\
-1 & +1 & -1 & -1 & -1 \\
-1 & -1 & +1 & -1 & -1 \\
-1 & -1 & -1 & +1 & -1 \\
-1 & -1 & -1 & -1 & +1
\end{pmatrix} \quad (6.80)
\]

With respect to the mapping matrix $M(k_1, k_2)$, the relabeling map becomes:

\[
\hat{y}_k^i = M(y_i, k) \quad (6.81)
\]

Suppose, there is a test sample ($x_s \in \mathbb{R}^f, y_s = 1$). Now, the classification of $x_s$ by the 5 SVMs yields a classification vector $\hat{y}_s = (1, -1, -1, -1, -1)^T$, i.e. the first SVM yields $+1$ and all other $-1$. One easily infers that the likelihood of $x_s$ being of class $k$ can be described by the “probability” vector:
\[
p(y_s) = M \cdot \hat{y}_s = \left( \begin{array}{c} \phi_1 \cdot \hat{y}_s \\ \vdots \\ \phi_C \cdot \hat{y}_s \end{array} \right) = \left( \begin{array}{ccccc} +1 & -1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & -1 \\ -1 & -1 & +1 & -1 & -1 \\ -1 & -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & -1 & +1 \end{array} \right) \cdot \left( \begin{array}{c} +1 \\ -1 \\ -1 \\ -1 \\ -1 \end{array} \right)
\]

(6.82)

\[
p = (5, 1, 1, 1, 1)^T
\]

(6.83)

According to the above formula, each of the \( C \) classes has a corresponding coding vector \( \phi_k \) which describes the optimal output of the SVM ensemble in case \( y_i = k \). The final voting in the OVA scheme, i.e. the real multi-class classification, is performed via max-voting:

\[
y_{OVA} = \arg \max (y_s)
\]

(6.84)

With respect to the example above, 5 SVMs voted for class one, therefore the argument with the highest value in the probability vector is \( y_{OVA} = 1 \). This scheme looks like a viable solution. Nonetheless, it comes with baggage. Assume a single SVM, the third e.g., voted “wrong”, i.e. \( \hat{y}_s = (+1, -1, +1, -1, -1)^T \). The probability vector becomes \( p(y_s) = (+3, -1, +3, -1, -1) \). In conclusion, a single result “flip” in a single SVM is enough to render the decision-making infeasible.

### 6.4.2 Error correcting output codes (ECOC)

The problem above has an intriguing correspondence in communication theory, which deals with a similar issue. Let us say Computer A wants to transfer information to computer B. Information is quite an abstract concept, that is why it has to be encoded into something a machine can understand. The letter “P” for example can be encoded using an ASCII-table into an 8-bit message: 01010000. The bit-stream can be transmitted via a copper line and received by computer B. B has to know the ASCII-table in order to decode the message back into the real information: letter “P”. This is just what one does in SVM ensemble classification: The information is the class-membership \( y_s \). This information is encoded into a sequence of +1 and -1 (the codes \( \phi_k \)) and transferred via the sample’s feature vector \( x_s \) to the SVM ensemble. The mapping matrix \( M(k_1, k_2) \) corresponds to the ASCII-table, which serves as a dictionary identifying the codes with the real information.

Both situations share the same problem: noise. The flip of bits of an 8-bit ASCII message is just as likely as a misclassification of a single SVM. Computer technology found a way to solve this issue via parity bits. One simply adds another checksum bit to the 8-bit message, which is one for an odd number of non-zero bits in the message and zero in all other cases. If the checksum check fails at computer B, the message is wrong and should be resent. Obviously, the number of parity bits can be easily increased. This raises security, but simultaneously expands the number of bits to be sent for a single message.

With regard to the SVM ensemble, one might add additional support vector machines that do not only investigate one class against all others. SVMs which are trained for separating two classes, say one and two, from the rest might be useful. As in the communication case, adding more “parity” SVMs can potentially increase error correction capabilities, but at the expense of higher computational complexity.

In conclusion, one has to find a compromise between fast and noise-sensitive ensembles and slow and
6.4 Multi-class support vector machines

noise-robust ones. This trade-off can be more easily understood in the framework of error correcting block codes. The matrix \( M \) is such a block code and its error detection/correction potential can be assessed calculating the minimal hamming distance \( d_{\text{min}} \) between the different coding vectors \( \phi_k \) of length \( L \). The hamming distance is defined as:

\[
d_H(\phi_{k_1}, \phi_{k_2}) = \sum_{j=1}^{L} |\phi_{k_1,j} - \phi_{k_2,j}| = \# \text{ of entries, differing in both vectors} \tag{6.85}
\]

and the minimal hamming distance of a block code \( M(k_1, k_2) \) is:

\[
d_{\text{min}}(M) = \min_{i \neq j} d_H(\phi_i, \phi_j) \tag{6.86}
\]

The minimal distance \( d_{\text{min}} \) is of great importance because the final classification outcome \( y_{\text{ECOC}} \) is determined by the minimal hamming distance of the SVM ensemble’s output \( \hat{y}_s \) towards all coding vectors:

\[
y_{\text{ECOC}} = \arg \min_k d_H(\phi_k, \hat{y}_s) \tag{6.87}
\]

Clearly, the higher the distance between the codes \( \phi_k \) in \( M \), the more misclassifications can be sustained. According to [265], the number of sustainable SVM "bit-flips" is given by \((d_{\text{min}}(M) - 1)/2\). The ground breaking idea by Dietterich [265] was to introduce techniques for creating error-correcting output codes (ECOC) to binary classifiers, just like the SVM. They revealed the correspondence between communication theory and SVM ensemble classification. Moreover, they determined a scheme for creating the most powerful coding matrix, given the number of classes \( C \) under investigation. This exhaustive code matrix can correct \( 2^{C-2} - 1 \) errors with the consequence of using \( 2^{C-1} - 1 \) SVMs. (see [265]) Given \( C = 13 \) classes, this would lead to the calculation of 4095 SVMs, which is nearly intractable for fast classification purposes. Therefore, they introduced several techniques for creating more advanced coding matrices \( M \) with great error-correcting capabilities: hill-climbing, a random greedy local search procedure (GSAT[266–268]) and a special class of codes: the Bose-Ray-Chaudhuri (BCH)-codes.[269] (see also [270]). The details of these procedures are spared here.

The major task for a practitioner is to select a suitable length \( L \) of the codes. The resulting \( C \times L \) matrix should be able to correct as many errors as possible, while keeping an eye on the computational complexity.

6.4.3 A unified approach

So far all schemes, the OVA and ECOC scheme, required using all samples within the training set, as all classes have been used for the SVM calculation. Clearly, less data demanding multi-class schemes exist. A popular example is the one vs. one scheme (OVO): Each SVM is trained using only samples from class \( i \) (labeled as 1) and \( j \) (labeled as -1). This reduces data consumption and computation time and is quite powerful. But how does this scheme relate to other multi-class schemes. Allwein et al. [209] proposed a framework from which all major ensemble strategies can be derived. The trick is to simply allow entries in \( M(k_1, k_2) \) to be zero. In this respect, the OVO scheme (for \( C = 5 \)) reads as:
The hamming distance with respect to the coding function $\phi_{k_1}$ and $\phi_{k_2}$ can be expressed by:

$$d_H = \frac{L - \phi_{k_1} \cdot \phi_{k_2}}{2}$$

(6.89)

With this scheme at hand one can summarize all multi-class strategies mentioned so far and assess their error-correcting capabilities (see [209]):

- one-vs-all (OVA: $\rho = 2C$): $M$ is a $C \times C$ matrix filled with $-1$, except the diagonal (filled with ones)
- one-vs-one (OVO: $\rho = \binom{C}{2} + 1$): $M$ is a $C \times \binom{C}{2}$ matrix filled with zeros, except the tuples $(i,j)$
- exhaustive dense ECOC ($\rho = 2^{C-2}$): $M$ is a $C \times 2^{C-1} - 1$ filled according to [265]
- dense ECOC ($\rho \leq \frac{L+1}{2}$) $M$ is a $C \times L$ matrix filled with random elements $-1, +1$
- sparse ECOC ($\rho \leq \frac{L+1}{2}$) $M$ is a $C \times L$ matrix filled with random elements $-1, 0, +1$

The matrices in the latter two schemes are created by choosing a sound length of the codes $L$. Afterwards, millions of random matrices of the size $C \times L$ filled with either $\{-1, +1\}$ (dense) or $\{-1, 0, +1\}$ (sparse) are created, seeking for the highest minimal hamming distance $\rho$. In the case that the final matrix $M$ exists, the ensemble classification is simple. One has to create $L$ different SVMs. Before the training starts, the multi-class labels $y_i$ have to be re-labeled into $\hat{y}_i$ according to eq. 6.81. The only difference now is that if a label is zero ($\hat{y}_i = 0$), this sample is not used during training. With regard to the final classification, any zero-entry in a code $\phi_k$ results in discarding the result of the corresponding SVM.

Are ECOC approaches superior? In summary, four basic SVM ensemble schemes are available that can all be understood within the framework of error-correcting output codes (ECOC). The critical reader, already, might have doubts if this generalization improves the former concepts: OVA and OVO. There is a potential flaw in comparing the noisy transmission channel in communication systems to the noisy feature vector $x_s$ classified by the SVM ensemble: the independence of errors. The key requirements for successful error correction are that errors are rare and that the same error does not manifest at many sites. A moderately noisy communication channel will easily meet these criteria. The SVM ensemble is quite different. This paragraph will discuss two major concerns: the independence of errors is broken because all SVMs are calculated on the same high-dimensional feature space consisting of the same samples. Furthermore, solely focusing on a high minimal hamming distance $\rho$ can lead to coding matrices $M$ that pose hardly solvable optimization problems. (see the brief summary of findings by Allwein et al. [209] below)

\footnote{In [209] $L = 10 \cdot \log_2(C)$ (dense) and $L = 15 \cdot \log_2(C)$ (sparse)}
6.4 Multi-class support vector machines

Suppose, there are $C = 13$ classes and the dense ECOC scheme with $L = 50$ SVMs is used to classify two samples $(x_1, y_1 = 1)$, $(x_2, y_2 = 1)$. The major problem now is the following: $x_1$ is at the edge of the hypersphere that comprises the majority of samples in class one. The sample $x_2$, on the other hand, is very close to the center of the hypersphere of samples in class two. Each SVM that seeks to separate samples of class one from other classes will inevitably fail in putting $x_2$ on the right side of the hyperplane. Hence, the error will propagate to all these SVMs and the independence of SVM errors is broken. The situation for $x_1$ is quite different. Chances are high that this sample will be at the right side of the different hyperplanes trained by SVMs, which try to separate the samples in class one from others. In case a certain SVM puts the sample on the "wrong" side, other SVMs will backup and correct those errors. Therefore, the performance boost by ECOC strategies is governed by the ratio between samples that are $x_1$ like and those that are $x_2$ like.

The second concern raised above was studied by Allwein et al. [209]. They give an upper bound for the training error $\epsilon_{ECOC}$ one can possibly obtain using Hamming decoding:

$$\epsilon_{ECOC} = \frac{2L \cdot \epsilon_{bin}}{\rho \cdot c_0^{Loss}} \rho^{vd/2} \approx 4 \cdot \frac{\epsilon_{bin}}{c_0^{Loss}}$$  \hspace{1cm} (6.90)

This bound depends on the average binary error $\epsilon_{bin}$, i.e. the average error obtained by the $L$ binary SVMs. Moreover, it takes into account the length of the codes $L$ and the minimal Hamming distance $\rho$ which can be at most $(L + 1)/2$. This formula states that despite a certain constant $c_0^{Loss}$, the training error is at most four times the average binary error $\epsilon_{bin}$. This holds only if the matrix $M$ is optimal, i.e. $\rho = (L + 1)/2$.

The major of finding of Allwein et al. [209] is the following: Even though exhaustive ECOCs provide higher error correction capabilities (high $\rho$), they exhibit increased average binary error $\epsilon$. In sum, the performance of the OVA scheme is still comparable to ECOC approaches. This is even more significant as they train much more SVMs. The major problem is that the binary problems posed by the ECOC matrices are much more difficult to train than those of the OVO and OVA schemes. The increased minimal hamming distance is traded for worse posed binary SVM problems.

6.4.4 Scheme selection

Even though this unified approach allows comparing ensemble strategies and gives an upper error bound, decision-making has not eased. A first study, which compared different multi-class schemes on several data sets was carried out by Hsu and Lin [264]. They claim that there is a significant difference between those strategies favoring the OVO scheme and an “all-together” strategy (DAG) by [272]. Rifkin and Klautau’s [213] efforts to bring the discussion to solid ground point towards another direction. They emphasize that even though there are relative differences between the best and worst scheme in the study of Hsu and Lin [264], the absolute differences are of minor significance. The question is: Why should one employ time-consuming algorithms if the difference ranges from 4 samples in 14500 to 20 in 2000.

Rifkin and Klautau [213] concludes that the results of Hsu and Lin [264] “support the notion that at least as far as accuracy is concerned, when well-tuned binary classifiers (in this case SVMs with

28 From a more abstract perspective, all those schemes relate closely to mixture of experts proposed by [30] and major work by [271]. Their many “experts” (neuronal networks) receive the feature input and the additional gating neuronal network, whose “opinion” (output of the neuronal network in question) should be listened to.
RBF kernels) are used as the underlying classifiers, a wide variety of multi-class schemes are essentially indistinguishable.”[213] Moreover, they show in their own analysis that ECOC strategies are not superior to the OVA scheme. At first glance, this is in contrast to the propositions of [265, 273, 274], which state, especially [274], that ECOC reduces bias and variance of the overall solution. At second glance, this is not surprising as their work primarily focused on decision tree algorithms like CART or C4.5. The major requisite for ECOCs to work is that the errors made by the binary induction algorithm are highly uncorrelated. [265] The decision trees used by [265, 273, 274] exhibit just that property. Hence, ECOCs are superior. Rifkin and Klautau [213] could show that in case the binary SVMs are optimally tuned by means of hyper-parameter selection, then the OVA scheme is just as good as ECOC approaches. Thus, the errors of optimally tuned SVMs are highly correlated, causing ECOC approaches to “fail”. 29

The results of Rifkin and Klautau [213] are based on the assumption that the results of the binary SVMs are combined using the minimum hamming distance approach. This is obviously not the only way. It speaks volumes that a recent extensive study by [15] uses nine different combination strategies for the OVO and two for the OVA scheme. A lot of statistically based approaches calculating posterior probabilities [275, 276] have been proposed. Nonetheless, Galar et al. [15] showed that the simple VOTE algorithm 30 for the OVO scheme is not beaten by any other combination rule on a wide range of datasets. In contrast to [213] though, they show that the OVO is significantly better than the OVA scheme. Yet, they admit to using not optimally tuned classifiers. Another disappointing fact about the study is the comparison between support vector machines and fuzzy support vector machines (see [277–279]). As the study compares many classifiers with their corresponding combination strategies, it would have been nice too see if SVMs are really inferior to FSVMs31. Unluckily, the parameterization of [15] shows that both use polynomial kernels of degree one. As no information about the corresponding α is given, one can assume that this is basically a linear, not well-tuned SVM. Hence, one might ask if the superiority of PDFC over SVMs is basically due to its more sophisticated intrinsic combination approach. Despite those minor remarks, this is an extensive study, worth reading for anyone diving into the world of machine learning. A striking property of it is that it introduces the κ-index for comparing classifiers. As already stated in the introduction, accuracy might be not the best judge for performance evaluation.

In conclusion no study has demonstrated that the OVO-scheme has inferior performance on a wide range of datasets. This is in accordance with other researchers in the field of ECG-classification employing this scheme [280]. The major leap of ECOC schemes is that they mostly randomly created. Allwein et al. [209] emphasized early-on that joining samples from different classes only due to an optimization independent from the underlying data structure is problematic. Thus, research on data dependent ECOC schemes gains momentum. An ECOC software library hosted by [281] provides extensive methods for ECOCs. The most prominent one is the ECOC-ONE algorithm by [282], which claims superiority over OVO. Other authors use heuristically determined schemes.[41]. Even though new approaches by Shiraiishi and Fukumizu [283], Zhang et al. [284] claim superior power by training the combiner of the SVMs, it was not included in this thesis. Instead, this thesis will compare the four schemes: OVO, OVA, ECOC(dense) and ECOC(sparse) using well tuned binary classifiers and test them on the datasets provided by Galar et al. [15].

6.5 The final setup

The final setup is the following: First, one splits the feature matrix according to their class-membership, which yields \( C = 13 \) sub-matrices containing the samples. Now, one has to create a training and test

29 Tsipouras et al. [34] back this claim by showing equality between the OVO and OVA scheme in their calculations. 30 which is essentially the hamming decoding discussed above 31 A study by [108] suggests that it is computationally more expensive
set. The hyper-parameter selection (optimization of the SVM) will be carried out on the training set. The final evaluation is done on the test set. A major concern in this respect is the training / test ratio \( r_{train} = \frac{N_{train}}{N_{train} + N_{test}} \). To gain a certain degree of comparability, the ratios for the different ECG-classes are chosen according to [17]:

\[
\begin{align*}
    r_{train}^N &= \frac{1}{4} \\
    r_{train}^{VES} &= r_{train}^{RBBB} = r_{train}^{LBBB} = r_{train}^1 = \frac{1}{3} \\
    r_{train}^{else} &= \frac{1}{2}
\end{align*}
\]

The rationale behind this choice is that the normal beats should be less dominant during training\(^{32}\). Moreover, testing a large amount of normal beats (75\%) assures a realistic estimate of the classification power. After splitting the data, one has to construct the binary classifiers. Thus, \( C(C-1)/2 \) binary SVMs are trained using M\(_{OVO}\)(\( C = 13 \)), i.e. all samples of the \( i^\text{th} \) class are labeled 1 and those of class \( j \) are labeled \(-1\). Each binary setup is now object to a cross-validated hyper-parameter search. For the pre-evaluations I use the faster two-fold cross validation (see subsection 6.2.1). In the final evaluations, five-fold cross validation is used. The grid search comprises the fast \( \gamma \)-determination proposed in this thesis using a range of \(-20\) to \(+20\) in 0.25 steps. The cost factor \( C \) is determined in a \(-8\) to \(2\) range in accordance with [27]. After the optimal configuration has been found, the SVM is trained on the whole training set using this very setup. During training each sample has to be tested by each binary classifier. The outcome of the SVMs is combined according to the last section.

\(^{32}\) [83] also emphasizes the necessity of weighting classes against domination of classes containing more samples
Chapter 7

Results

The last chapters provide new powerful tools for feature extraction, feature selection and fast hyperparameter search for support vector machines. This chapter will present the experiments that have been necessary to verify my key findings. The results will be presented in reverse order, i.e. first, findings with regard to SVMs and at the end, the $\rho$-Score review is shown. This is due to the fact that the support vector machine setup has to be fixed first. Afterwards, one can evaluate the performance of the feature selection, feature extraction techniques and the final ECG-classifier setup.

The procedures carried out in this chapter are summarized in fig. 7.1. First, preprocessing and feature extraction have been conducted to generate the feature database (see section 4.8). It consists of 1.4 million features generated for each of the 100,000 beats under investigation. The second step is to fix the SVM setup, i.e. determine the hyper-parameter selection strategy, the normalization and multi-class methods to be employed. Therefore, two databases, the Galar[15] and Rätschlab[29] Machine-learning database (MLDB), have been utilized to question our main propositions towards support vector machines (see section 2.2.4 for database characteristics):

- Hyper-parameter selection can be fastened using histogram-based kernel optimization without loss of performance
- Normalization techniques have a minor impact on SVM performance (also low dependency on data used)
- Dense and sparse ECOC matrices show high performance stability, when facing low data amounts
- Opposed to [209], the Hamming decoding is more powerful than loss decoding

The analysis of the constraints of SVM use will yield two setups: a high performance and a fast one. The latter will be further used for the evaluation of the feature selection schemes. In chapter five, 12 different effect-size indices have been introduced: standard ones (see p. 88) and the novel dichotomized effect-size indices (DESI, see p. 94). These are employed to calculate the inter-class scoring table (IST), which is subsequently exploited to generate the feature sets. The generation process can be carried out via four strategies: best per class (TopPC), highest mean IST (TopMean), IST table filling (TableFill), decorrelated best per class (TopPCCorr). One option would be to use the full feature DB and calculate the best combination and parameterization for each effect-size / strategy tuple. The major problem is that the data used for the evaluation of the best feature selection strategy is also used to generate the final ECG-classifier. Therefore, "training" the feature selector to the data at hand should be omitted. Hence, only one tenth of all features in the feature DB will be used to assess the optimal selection method. Moreover, the effect-size index will be determined by the evaluation of the TopPC and TopMean strategies only. Afterwards, the best index is tested on the remaining four strategies. For more details
Chapter 7 Results

1. Data Generation
MIT BIH Preprocessing Feature Extraction Feature-DB

2. SVM-setup determination
Galar MLDB Hyper-parameter selection Multi-class methods Norm and size dependence

3. Feature selection setup
Fast SVM setup OVO/Hamming, Full Search Hashing grid search 10% of Feature-DB Calc interclass tables

4. Full Test
Top Mean resp. Top PC Feature Selection Use best setup on full Feature-DB SVM-Eval

5. Further Analysis
Full Feature-DB Feature Extraction MIT ST Longterm DB Test best feature extractors Test best ECG normalizers

Full Review of R-Score

The myth of the perfect wavelet

Figure 7.1: The major scheme for the result chapter. The blue fields denote the datasets used for the connected tasks. The red rectangles marks the evaluations performed and the green ones show calculations carried out on the servers.
For the final feature-set evaluation, the best strategy will be carried out using the full feature DB. The best of the 10 training-test splits will be selected as the final feature set. Afterwards, the performance of this set is assessed using 10 new training-test splits. The ultimate test is to evaluate this set on a completely different database, the ECG longterm DB.

At the end of this chapter I will show which feature extraction strategy is more powerful. Moreover, experiments are carried out which reveal that "the perfect wavelet" does not exist. The chapter will conclude with showing the performance of the proposed feature set and ECG-classifier within a review of 72 modern ECG-classifiers.

### 7.1 SVM setup determination

As can be seen from the scheme in fig. 7.1, the SVM setup has several constraints. First of all, the computation of the binary SVMs depends on the hyper-parameter selection strategy, the normalization employed, the data used and the kernel to be utilized. The combination of those binary SVMs, generating a multi-class SVM, can be guided by strategies like one vs. one (OVO), one vs. all (OVA) and error correction output codes (ECOCs), which can be Dense or Sparse. As the unified multi-class scheme by Allwein et al. [209] is used, those four schemes reduce to ECOC schemes of different shape and performance. Another advantage of this unification is that one can test two different decoding approaches one each of the four schemes: the Hamming and Loss-Decoding.

This section will support the following assumptions:

- Histogram optimization is a very fast and high-performing alternative for selecting hyper-parameters in binary SVMs
- Histogram optimization can decrease computation time in multi-class setups tremendously, while showing no loss in OVA and OVO and only slight performance losses (1 or 2 percent) in ECOC schemes.
- Hamming decoding is superior over Loss decoding.  
- OVA and OVO depend on the combination scheme. (Loss and Hamming Decoding)  
- ECOC is superior over OVO which is superior over OVA.  
- ECOC strategies are more stable when facing low amounts of data.
- The majority of normalization strategies yield equal performance.
- The amount of data used for normalization estimates is of insignificant importance.

Some remarks about the data which was used to provide evidence in favor or disadvantage of the above propositions can be found in subsection 2.2.4. In the next subsections, the performance of different

---

1 This is opposed to findings of Allwein et al. [209], who have not used SVM optimization and did not use inference testing to support their claim.  
2 This is opposed to findings by Galar et al. [15], who have not used SVM optimization and used polynomial kernels  
3 This is in accordance with [15, 265]
multi-class schemes, normalizations and hyper-parameter selection strategies are compared. Thus, the Wilcoxon signed-rank test for paired samples is employed. This assures that the measured differences are statistically significant. The level of significance is chosen to be $\alpha = 0.05$, which is a more strict choice than $\alpha = 0.10$ in [15]. The choice of a non-parametric method follows according to [15].

### 7.1.1 A faster hyper-parameter selection

Now, let us check if the proposed histogram hyper-parameter selection is as good as a Full-Grid search[210]. It is clearly faster, as it runs less cross-validations. Two versions of the histogram-based search were tested. The basic procedure, as described in section 6.3.4, is denoted as "Histo". As can be seen in tab. 7.1, for specific datasets (Splice) the kernel width $\gamma_{Hist}$ is not as accurate as possible and therefore the final hyper-parameter pair $(C_{opt}, \gamma_{Hist})$ yields a lower performance. Hence, the "HistoLarge" approach assesses the pairs $(C_{opt}, 2^{\log(\gamma_{Hist})+\Delta}) | \Delta \in \{-1, +1, +2\}$ to check the performance of slightly higher or lower $\gamma$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>0.1235</td>
<td>0.1178</td>
<td>0.1121 ± 0.0054</td>
<td>0.1161</td>
<td>0.1161</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>0.0248</td>
<td>0.0248</td>
<td>0.0246 ± 0.0025</td>
<td>0.0257</td>
<td>0.0218</td>
<td></td>
</tr>
<tr>
<td>Splice</td>
<td>0.0970</td>
<td>0.1011</td>
<td>0.1017 ± 0.0065</td>
<td>0.0970</td>
<td>0.2382</td>
<td></td>
</tr>
<tr>
<td>Waveform</td>
<td>0.1078</td>
<td>0.1078</td>
<td>0.1120 ± 0.0044</td>
<td>0.1117</td>
<td>0.1143</td>
<td></td>
</tr>
<tr>
<td>diabetes</td>
<td>0.2433</td>
<td>0.2433</td>
<td></td>
<td>0.2400</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>ringnorm</td>
<td>0.01429</td>
<td>0.0148</td>
<td></td>
<td>0.0167</td>
<td>0.017</td>
<td></td>
</tr>
<tr>
<td>twonorm</td>
<td>0.031</td>
<td>0.02914</td>
<td></td>
<td>0.0271</td>
<td>0.0271</td>
<td></td>
</tr>
<tr>
<td>No. CV</td>
<td>441</td>
<td>54</td>
<td>13</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Accuracy of five different hyper-parameter selection strategies for seven UCI datasets: Standard grid-search by [210], improved search by [26], uniform designs by Huang et al. [27] and two versions of the parameter search proposed in this thesis. The number of needed cross-validation steps is noted below the author’s name.

Table 7.1 presents the results for the binary datasets described in the last subsection. From tab. 7.1 one easily deduces that there is no statistically significant ($p < 0.05$) difference between our histogram approximation and the grid search. On some datasets it performs better (banana, diabetes, twonorm), on some worse (image, waveform, ringnorm), on some it is just the same (splice). This demonstrates that the hyper-parameter selection strategy proposed in this thesis is a viable solution. It also uses less cross-validation than all other compared hyper-parameter selection schemes.

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Accuracy</th>
<th>$\kappa$</th>
<th>$T$</th>
<th>$S$</th>
<th>PPW</th>
<th>NPW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid vs. Histo</td>
<td>n.s.</td>
<td>0.0126 ± 0.0251 *</td>
<td>0.0260 ± 0.0434 *</td>
<td>0.0025 ± 0.0043 *</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>Grid vs. Histo + UD</td>
<td>n.s.</td>
<td>0.0107 ± 0.0254 *</td>
<td>0.0226 ± 0.0423 *</td>
<td>0.0023 ± 0.0042 *</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
</tbody>
</table>

Table 7.2: Showing the difference between optimization schemes using OVO with Hamming decoding: Sensitivity, Kappa-index and Specificity drop slightly in histogram optimization. No significant change between the histogram schemes. Moreover, PPW, NPW and Accuracy remain unchanged. Significant changes ($p < 0.05$) are indicated by *

Clearly, showing this on just a few binary datasets is not enough. The multi-class scenario, wherein one must carry out full cross-validation on several binary problems, is a pretty good assessment for a hyper-parameter selector. In there, one has to carry out a full cross-validation on several binary problems. Therefore, an inferior choice of hyper-parameters will yield reduced performance. According to the results on the multi-class database in tab. 7.3, the optimization strategies show slight differences in accuracies. So how significant are they? For this purpose, Wilcoxon’s signed-rank test for paired samples was carried out. It shows that there is no statistically significant difference in accuracy comparing the three schemes. The $\kappa$-index though, is slightly higher using the grid optimization. This is due to a
lowered mean-per-class sensitivity ($T$) and specificity ($S$). The predictive values, both the negative and positive one, show no statistically significant difference. These findings are summarized in tab. 7.2 and hold for the OVO and OVA scheme. In a later subsection, one can see that the optimization scheme lowers $\kappa$-index by one or two percent points, given Dense or Sparse ECOCs. This is in accordance with the finding that the lack of hyperparameter optimization in Galar et al. [15] and Allwein et al. [209] can cause severe misinterpretations of the results at hand.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Strategy</th>
<th>Accuracy</th>
<th>Kappa-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>autos</td>
<td>FullGrid</td>
<td>0.7547</td>
<td>0.6797</td>
</tr>
<tr>
<td></td>
<td>Histo</td>
<td>0.7421</td>
<td>0.6618</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>0.7484</td>
<td>0.6706</td>
</tr>
<tr>
<td>car</td>
<td></td>
<td>0.9925</td>
<td>0.9836</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9855</td>
<td>0.9684</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9850</td>
<td>0.9671</td>
</tr>
<tr>
<td>cleveland</td>
<td></td>
<td>0.5791</td>
<td>0.3122</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5219</td>
<td>0.2280</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5421</td>
<td>0.2516</td>
</tr>
<tr>
<td>dermatology</td>
<td></td>
<td>0.9693</td>
<td>0.9615</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9693</td>
<td>0.9615</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9693</td>
<td>0.9615</td>
</tr>
<tr>
<td>ecoli</td>
<td></td>
<td>0.8214</td>
<td>0.7526</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8185</td>
<td>0.7466</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8333</td>
<td>0.7685</td>
</tr>
<tr>
<td>flare</td>
<td></td>
<td>0.7223</td>
<td>0.6415</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.7280</td>
<td>0.6482</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.7308</td>
<td>0.6522</td>
</tr>
<tr>
<td>glass</td>
<td></td>
<td>0.7290</td>
<td>0.6301</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.6822</td>
<td>0.5623</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.6682</td>
<td>0.5430</td>
</tr>
<tr>
<td>led7digit</td>
<td></td>
<td>0.7320</td>
<td>0.7020</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.7400</td>
<td>0.7107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.7280</td>
<td>0.6974</td>
</tr>
<tr>
<td>lymphography</td>
<td></td>
<td>0.8176</td>
<td>0.6416</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8176</td>
<td>0.6366</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8108</td>
<td>0.6252</td>
</tr>
<tr>
<td>nursery</td>
<td></td>
<td>0.8896</td>
<td>0.8379</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8772</td>
<td>0.8195</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8726</td>
<td>0.8127</td>
</tr>
<tr>
<td>pageblocks</td>
<td></td>
<td>0.9471</td>
<td>0.7088</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9453</td>
<td>0.6958</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9471</td>
<td>0.7030</td>
</tr>
<tr>
<td>penbased</td>
<td></td>
<td>0.9718</td>
<td>0.9687</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9745</td>
<td>0.9717</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9755</td>
<td>0.9727</td>
</tr>
<tr>
<td>satimage</td>
<td></td>
<td>0.8896</td>
<td>0.8632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8869</td>
<td>0.8631</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8771</td>
<td>0.8478</td>
</tr>
<tr>
<td>segment</td>
<td></td>
<td>0.9701</td>
<td>0.9652</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9645</td>
<td>0.9586</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9658</td>
<td>0.9601</td>
</tr>
<tr>
<td>shuttle</td>
<td></td>
<td>0.9963</td>
<td>0.9897</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9922</td>
<td>0.9782</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9917</td>
<td>0.9770</td>
</tr>
<tr>
<td>vehicle</td>
<td></td>
<td>0.8499</td>
<td>0.7998</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8345</td>
<td>0.7794</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8404</td>
<td>0.7872</td>
</tr>
<tr>
<td>vowel</td>
<td></td>
<td>0.9838</td>
<td>0.9822</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9798</td>
<td>0.9778</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9798</td>
<td>0.9778</td>
</tr>
<tr>
<td>zoo</td>
<td></td>
<td>0.9307</td>
<td>0.9085</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9505</td>
<td>0.9344</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9604</td>
<td>0.9475</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>0.8637</td>
<td>0.7960</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td></td>
<td>0.8563</td>
<td>0.7835</td>
</tr>
<tr>
<td>No. CV</td>
<td></td>
<td>0.8570</td>
<td>0.7853</td>
</tr>
<tr>
<td>441</td>
<td>10</td>
<td>7</td>
<td>441</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 7.3: Accuracy of 3 different hyper-parameter selection strategies for 18 UCI datasets using OVO with Hamming decoding: Standard grid-search by [210] and two versions of the parameter search proposed in this thesis. The number of needed cross-validation steps is noted below the author's name. The left resp. right side shows the accuracy resp. the $\kappa$-index for each method for every dataset.

7.1.2 Data normalization

The normalization of the data is vital for SVMs to generate proper models. Nonetheless, little is known about the effects of different normalization strategies on the final outcome of the classifier, especially the SVM. The basic recommendation is to normalize the data within a range of $[-1, 1]$ or $[0, 1]$. The purpose of normalization is to ensure that features with smaller values are rendered insignificant by features of higher values. Moreover, this is advantageous for the numerical stability of the solution. A major problem arises from the fact that the normalization constants need to be calculated only during training in order to avoid over-fitting. Suppose the training sample $x^{train}_i$ is normalized using MinMax-normalization:

$$x^{\hat{}}_{train} = \frac{x^{train}_i - \min(x^{train})}{\max(x^{train}) - \min(x^{train})}$$ (7.1)

The resulting normalized feature $\hat{x}^i$ is within the range $[0, 1]$. Unfortunately, this assumption does not necessarily hold for the test samples in $x_{test}$. The range of the normalized feature remains fuzzy,
as $\min(\bar{x}_{\text{train}})$ is only a sample estimate of the true $\min(\bar{x})$. Therefore, one has to investigate, if this fuzziness has a strong impact on the classification performance. The box plots in fig. 7.2 demonstrate the impact of normalization on the datasets of Galar et al. [15] with respect to the eleven normalizations introduced on p.122.

![Box plots showing accuracy relative to the normalization with the highest mean accuracy: GSig for OVO (top), Std for SPARSE (bottom); for all eleven normalizations sorted with respect to their mean accuracy in descending order. The black, blue, green and red dotted rectangles comprise normalizations, which show no significant difference.](image)

Obviously, there is a group of six normalizations which show indistinguishable results on both combination rules. The GSig, Gauss, Std, Rank, Max and Minmax norm are superior to other normalization techniques, especially the Raw norm, which is essentially not doing normalization at all. From this result one might infer, that even though normalization does improve results, there is no silver bullet. It
is up to further research whether or not strategies can be determined which identify the most suiting normalization for the dataset at hand.

### 7.1.3 Combiners

In the last two subsections I shed light on the notions of hyper-parameter selection and normalization. Both have a strong impact on the outcome of the binary SVMs. In the multi-class scenario, one has to investigate the technique for combining the well-tuned binary SVMs. Therefore, this subsection is going to analyze four combination schemes: the one against all (OVA), one-vs.-one (OVO) and error correcting output codes (ECOCs) either using Dense or Sparse codes. So, the first question is how do those four schemes perform on the multi-class datasets provided by Galar et al. [15]?

Table 7.4: The mean and standard deviation of the accuracy, $\kappa$-index, sensitivity, specificity, positive/negative predictive value over 18 datasets provided by [15] for four different multi-class strategies: Dense-ECOCs, Sparse-ECOCs, OVO and OVA.

<table>
<thead>
<tr>
<th></th>
<th>Dense</th>
<th>Sparse</th>
<th>OVO</th>
<th>OVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc</td>
<td>label 87.38 ± 11.34</td>
<td>87.05 ± 11.90</td>
<td>86.39 ± 12.00</td>
<td>83.69 ± 13.78</td>
</tr>
<tr>
<td></td>
<td>prob 87.13 ± 11.66</td>
<td>87.21 ± 11.67</td>
<td>83.94 ± 11.59</td>
<td>4.56 ± 6.97</td>
</tr>
<tr>
<td>$\kappa$-index</td>
<td>label 81.15 ± 16.83</td>
<td>80.46 ± 17.84</td>
<td>79.41 ± 18.22</td>
<td>74.32 ± 23.74</td>
</tr>
<tr>
<td></td>
<td>prob 80.80 ± 17.45</td>
<td>80.74 ± 17.61</td>
<td>76.77 ± 16.70</td>
<td>12.63 ± 9.54</td>
</tr>
<tr>
<td>Sens.</td>
<td>label 77.22 ± 17.50</td>
<td>76.90 ± 18.54</td>
<td>75.28 ± 19.00</td>
<td>72.34 ± 20.72</td>
</tr>
<tr>
<td></td>
<td>prob 77.58 ± 17.57</td>
<td>77.25 ± 17.97</td>
<td>72.98 ± 17.00</td>
<td>3.34 ± 6.64</td>
</tr>
<tr>
<td>Spec.</td>
<td>label 96.62 ± 3.19</td>
<td>96.45 ± 3.40</td>
<td>96.30 ± 3.48</td>
<td>95.24 ± 4.74</td>
</tr>
<tr>
<td></td>
<td>prob 96.55 ± 3.31</td>
<td>96.54 ± 3.34</td>
<td>95.99 ± 3.27</td>
<td>80.64 ± 7.15</td>
</tr>
<tr>
<td>PPW</td>
<td>label 79.28 ± 17.02</td>
<td>79.51 ± 17.57</td>
<td>77.96 ± 18.28</td>
<td>77.47 ± 20.78</td>
</tr>
<tr>
<td></td>
<td>prob 79.14 ± 17.16</td>
<td>78.95 ± 17.67</td>
<td>76.22 ± 16.26</td>
<td>11.43 ± 10.28</td>
</tr>
<tr>
<td>NPW</td>
<td>label 96.94 ± 2.72</td>
<td>96.87 ± 2.81</td>
<td>96.71 ± 2.93</td>
<td>96.24 ± 3.47</td>
</tr>
<tr>
<td></td>
<td>prob 96.90 ± 2.70</td>
<td>96.94 ± 2.71</td>
<td>96.34 ± 2.90</td>
<td>79.59 ± 7.39</td>
</tr>
</tbody>
</table>

The table above demonstrates that in the case where a full grid search is performed, ECOCs are superior to the OVO or OVA scheme. This holds for all performance measures. One might wonder why standard deviation is so high. This is due to averaging over all datasets, which entails averaging over datasets of lower and higher basic performance. Hence, one is also interested in the mean change per dataset due to a change of the multi-class scheme. The table below provides extensive insight into the mean difference per dataset and the statistical significance of those differences.

The values on the diagonal show that except for the OVA scheme, there is no significant difference between loss-based decoding and Hamming decoding, though loss-based decoding tends to be slightly worse. Moreover, it is impractical to use loss-decoding in the OVA scenario. Hence, all indices concerning loss-based OVA are worse compared to the other multi-class schemes. Yet, even with regard to Hamming decoding, OVA is inferior to all other schemes with $p<0.001$ by means of accuracy and $kappa$-index. The difference in the latter is even more significant (around three percent points accuracy and six percent points kappa-index drop). One can infer from the other four measures that the major loss occurs due to reduced sensitivity. The positive predictive value does not even change significantly.

Let us look at the other 3 schemes. According to the table above, the dense and sparse matrices for the Allwein-scheme perform alike. There are no significant changes in any performance parameter. Yet both share the superiority over OVO multi-classification. OVO is much more inferior using loss-based decoding. Referring to the results of Galar et al. [15], the optimization of the Gaussian kernel and the cost parameter $C$ have a tremendously impact on the performance: Given a polynomial kernel with fixed
Table 7.5: The mean and standard deviation of the difference between combiner methods on a per dataset basis for six performance measures: Accuracy, \(\kappa\)-index, sensitivity, specificity, positive and negative predictive value (all values are percentages). The upper rows show an example denoting the purpose of each table entry. Each measure (including the example one) corresponds to a suitable. The upper triangular matrix within this subtable shows the difference between Sparse ECOC, Dense ECOC, OVO and OVA multi-class schemes using Hamming decoding by means of the respective performance assessor. The lower triangular matrix shows these differences with schemes using Loss decoding. On the diagonal, one can see the differences between Loss and Hamming decoding for the Sparse ECOC, Dense ECOC, OVO and OVA multi-class scheme. Example: The difference between the accuracy of a dataset assessed by the Sparse ECOC scheme (Loss decoding) and the accuracy using Hamming decoding averaged over all 18 datasets in [15] is \(-0.24 \pm 0.82\) percent points.
7.1 SVM setup determination

C, the κ-Index for OVO is $0.7243 \pm 0.1767$ [15]. Using a full grid search employing the Gauss kernel yields a κ-Index of $0.7941 \pm 18.22$. (see tab. 7.4)

Now, all the above measures have been carried out using the 21 x 21 grid from [210]. Hence, the question arises: Do values change tremendously when 'Histo' search is active? Moreover, one wants to investigate the impact of random grid parameters.

Table 7.6 clearly shows that random selection of kernel parameters yields a tremendous loss in accuracy. None of the 8 schemes presented here, shows a significant superiority even though dense ECOC and OVA (Hamming decoding) strategies can cope best with the suboptimal SVMs.

Moreover, 'Histolarge' shows nearly no change compared to the grid case. Only the accuracy for Dense ECOCs using Hamming decoding is slightly lower. Therefore, 'Histolarge' achieves high accuracies faster. The case for 'Histo' is not that impressive. One can see that there are no significant changes in the OVO and OVA situation. One can moreover see, that the higher performance of OVO vs. ECOC techniques necessitates highly tuned binary classifiers. If not they are still the same as OVO.

Size dependency  All information provided so far is based on a large amount of training data. Eighty percent of the data was used during each of the five test-training splits. It is of great interest to investigate how the combination methods perform when facing small amounts of data. For this purpose the following experiment has been conducted: The data has been split into a training and test set using a fixed ratio $r = \frac{N_{\text{train}}}{N_{\text{train}} + N_{\text{test}}}$. Galar et al. [15] used $r = 0.8$. The performance of the eight combiner schemes (Dense, Sparse, OVO, OVA with Hamming resp. Loss decoding) has been calculated for 10 different splits using $r \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.6, 0.7\}$ for all 18 datasets. The accuracy at $r = 0.7$, averaged over 10 splits, has been set as the reference point for each dataset. The difference between the accuracy at lower $r$ and the reference point averaged over all 18 datasets is depicted below for Dense ECOCs with Hamming Decoding. It shows the median and the upper/lower quartile (error-bars) of the accuracy difference with regard to the 18 datasets.

The plot above also reveals that the exponential function in eq. 7.2 nicely fits the experimental data. (see Adj. R-Square in tab. 7.7)

$$y = y_0 + Ae^{-\tau}$$

Consequently, this fitting procedure was applied to all eight combination methods (4 combiner schemes times 2 decoding schemes). Table 7.7 shows the minimal difference (MinDiff), the maximal difference (MaxDiff) and the ratio $r_{\text{opt}}$ for which the difference is only $y = -0.01$. $r_{\text{opt}}$ and its standard deviation (Gauss propagation of uncertainty) are calculated according to:

$$r_{\text{opt}}(y = -0.01) = \tau \ln \frac{A}{y - y_0}$$

$$\Delta r_{\text{opt}}(y = -0.01) = \sqrt{\Delta \tau^2 \left( \ln \frac{A}{y - y_0} \right)^2 + \tau^2 \left( \frac{\Delta A}{\Delta \tau} \right)^2 + \left( \frac{\Delta y_0}{y - y_0} \right)^2}$$

4 The standard deviation 0.0559 given in [15] is the average standard deviation of the cross-validation over all datasets. Other performance measures treated accordingly.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OVA</td>
<td>Hamm.</td>
<td>83.69</td>
<td>-0.14 ± 0.71</td>
<td>0.77 ± 1.84</td>
<td>34.47 ± 22.31***</td>
</tr>
<tr>
<td>Acc</td>
<td>Loss</td>
<td>4.56 ± 6.97</td>
<td>0.30 ± 2.40</td>
<td>0.80 ± 4.12</td>
<td>-6.16 ± 11.94*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVO</td>
<td>Hamm.</td>
<td>86.39</td>
<td>0.38 ± 1.34</td>
<td>0.60 ± 1.51</td>
<td>25.08 ± 15.05***</td>
</tr>
<tr>
<td></td>
<td>Loss</td>
<td>83.94 ± 11.59</td>
<td>-1.62 ± 5.37</td>
<td>-1.25 ± 5.43</td>
<td>23.95 ± 13.73***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sparse</td>
<td>Hamm.</td>
<td>87.38</td>
<td>0.93 ± 1.77</td>
<td>1.29 ± 1.88*</td>
<td>24.01 ± 15.06***</td>
</tr>
<tr>
<td></td>
<td>Sparse</td>
<td>Loss</td>
<td>87.13</td>
<td>0.36 ± 0.92</td>
<td>1.11 ± 1.68*</td>
<td>25.70 ± 16.38***</td>
</tr>
<tr>
<td></td>
<td>Dense</td>
<td>Hamm.</td>
<td>87.05</td>
<td>0.33 ± 0.67</td>
<td>1.05 ± 1.12***</td>
<td>27.62 ± 17.50***</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loss</td>
<td>87.21</td>
<td>0.43 ± 1.37</td>
<td>1.61 ± 2.21**</td>
<td>30.56 ± 17.44***</td>
</tr>
</tbody>
</table>

| Acc   | OVA    | Hamm.    | 74.32 | -0.29 ± 1.20     | 1.12 ± 2.32| 51.74 ± 24.75*** |
|       | Loss   | 12.63 ± 9.54 | -0.73 ± 1.60 | -1.28 ± 2.51 | 2.48 ± 6.76 |
| k-index| OVO    | Hamm.    | 79.41 | 0.76 ± 2.25      | 0.95 ± 2.61| 41.33 ± 19.34*** |
|       | Loss   | 76.77 ± 16.70 | -1.48 ± 6.30 | -0.90 ± 6.43 | 41.11 ± 20.68*** |
|       | Sparse | Hamm.    | 81.15 | 1.74 ± 3.01      | 2.33 ± 3.23| 37.62 ± 17.07*** |
|       | Sparse | Loss     | 80.80 | 0.95 ± 2.07      | 2.18 ± 3.10**| 42.97 ± 20.58*** |
|       | Dense  | Hamm.    | 80.46 | 0.70 ± 1.33      | 1.84 ± 1.70***| 44.83 ± 19.44*** |
|       |     | Loss     | 80.74 | 0.87 ± 2.44      | 2.81 ± 3.63**| 48.52 ± 16.32*** |

| Sens. | OVA    | Hamm.    | 72.34 | 1.20 ± 4.24      | 1.80 ± 4.48| 39.11 ± 21.73*** |
|       | Loss   | 3.34 ± 6.64 | -0.67 ± 2.69 | -0.78 ± 3.70 | -5.18 ± 5.63*** |
|       | OVO    | Hamm.    | 75.28 | 1.60 ± 3.78      | 1.82 ± 3.83| 31.66 ± 16.37*** |
|       | Loss   | 72.98 ± 17.50 | -0.20 ± 7.26 | -0.03 ± 7.04 | 31.08 ± 17.81*** |
|       | Sparse | Hamm.    | 77.22 | 2.45 ± 5.05      | 3.16 ± 5.30**| 30.18 ± 15.03*** |
|       | Sparse | Loss     | 75.88 | 2.26 ± 5.89      | 2.68 ± 6.12**| 33.53 ± 17.49*** |
|       | Dense  | Hamm.    | 76.90 | 1.70 ± 3.31**    | 2.75 ± 4.17***| 35.36 ± 18.27*** |
|       |     | Loss     | 77.25 | 1.66 ± 6.13      | 3.71 ± 6.29**| 39.15 ± 16.18*** |

| Spec. | OVA    | Hamm.    | 95.24 | -0.09 ± 0.27     | 0.21 ± 0.38| 9.26 ± 5.61*** |
|       | Loss   | 80.64 ± 7.15 | 0.23 ± 0.45 | 0.49 ± 0.87* | -1.49 ± 2.57*** |
|       | OVO    | Hamm.    | 96.30 | 0.20 ± 0.43      | 0.22 ± 0.54| 7.72 ± 4.75*** |
|       | Loss   | 95.99 ± 3.27 | -0.13 ± 0.76 | -0.00 ± 0.81 | 7.96 ± 5.37*** |
|       | Sparse | Hamm.    | 96.62 | 0.34 ± 0.57      | 0.44 ± 0.65**| 6.86 ± 3.42*** |
|       | Sparse | Loss     | 96.55 | 0.22 ± 0.47      | 0.47 ± 0.69**| 7.87 ± 4.33*** |
|       | Dense  | Hamm.    | 96.45 | 0.12 ± 0.27      | 0.32 ± 0.32***| 8.17 ± 4.39*** |
|       |     | Loss     | 96.54 | 0.23 ± 0.49      | 0.56 ± 0.76**| 8.83 ± 3.62*** |

| PPW   | OVA    | Hamm.    | 77.47 | 2.27 ± 6.53      | 3.50 ± 6.52*| 19.36 ± 23.55*** |
|       | Loss   | 11.43 ± 10.28 | 0.84 ± 7.48 | 1.17 ± 3.41 | -2.90 ± 6.57*** |
|       | OVO    | Hamm.    | 77.96 | 2.17 ± 7.53      | 2.13 ± 8.56**| 13.37 ± 13.85** |
|       | Loss   | 76.22 ± 16.26 | 0.91 ± 9.85 | 1.02 ± 9.87* | 16.18 ± 16.59** |
|       | Sparse | Hamm.    | 79.28 | 2.38 ± 6.69      | 4.16 ± 7.29**| 12.57 ± 13.81** |
|       | Sparse | Loss     | 79.14 | 2.15 ± 6.32      | 2.75 ± 7.00*| 14.53 ± 17.40** |
|       | Dense  | Hamm.    | 79.51 | 2.80 ± 5.94**    | 4.09 ± 6.73**| 14.58 ± 12.65*** |
|       |     | Loss     | 78.95 | 1.61 ± 6.36      | 3.42 ± 6.83*| 16.43 ± 10.61*** |

| NPW   | OVA    | Hamm.    | 96.24 | -0.20 ± 1.01     | 1.15 ± 3.61*| 9.19 ± 7.58*** |
|       | Loss   | 79.59 ± 7.39 | -0.03 ± 1.32 | 0.02 ± 1.39 | -1.36 ± 1.92* |
|       | OVO    | Hamm.    | 96.71 | 0.18 ± 0.53      | 0.21 ± 0.50| 4.72 ± 2.41*** |
|       | Loss   | 96.34 | -0.10 ± 0.74     | 0.04 ± 0.76| 5.73 ± 6.37*** |
|       | Sparse | Hamm.    | 96.94 | 0.20 ± 0.41      | 0.28 ± 0.57**| 4.27 ± 2.62*** |
|       | Sparse | Loss     | 96.90 | 0.04 ± 0.18      | 0.30 ± 0.56**| 5.58 ± 4.28*** |
|       | Dense  | Hamm.    | 96.87 | 0.08 ± 0.16      | 0.25 ± 0.29***| 4.67 ± 3.08*** |
|       |     | Loss     | 96.94 | 0.06 ± 0.32      | 0.43 ± 0.60**| 5.49 ± 2.70*** |

Table 7.6: The mean per dataset differences between kernel parameter selection by grid search, histologare, histo and random selection for six performance measures: Accuracy, k-index, specificity, sensitivity, positive and negative predictive value (all values are percentages). Each row contains: the performance measure, the multi-class scheme (Sparse, Dense, OVO or OVA), the decoding stage (Hamming or Loss-based), the performance measure averaged over all datasets using grid search, the difference between histologare, histo and random selection with respect to the grid search value on a per dataset basis.
7.1 SVM setup determination

The OVA scheme with Loss decoding was excluded from further analysis, as this scheme is already of low performance and the size dependency is negligible. A major difference can be seen between Hamming and Loss decoding. Hamming decoding reaches satisfactory accuracy levels earlier ($r_{opt}$ is smaller) and the accuracy is at most 11 or 12 percent points lower (see MaxDiff) than values for $r = 0.7$. The Dense-ECOC-scheme even reaches a maximum accuracy-drop of 8.53 ± 0.92 percent points on dataset-average (18 datasets). The OVO-scheme with Loss decoding forms the other end of the performance spectra. The drop there is 46.02 ± 3.31 percent points on average. Given $r_{opt} = 0.5 ± 0.05$ this scheme needs 39% ± 27% more data than the Dense-ECOC-scheme. For some datasets this could be a huge difference. One can conclude that the Hamming decoding leads to much more accurate solutions when facing less data. The superiority of Hamming over Loss decoding is in clear contrast to the results of citeallwein:2000. This stresses the notion that hyper-parameter search and optimal configuration of the SVM is crucial for a viable classification.

<table>
<thead>
<tr>
<th>Combiner</th>
<th>Decoding</th>
<th>Adj. R-Square</th>
<th>MinDiff [%]</th>
<th>MaxDiff [%]</th>
<th>$r_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense</td>
<td>Hamm</td>
<td>0.964</td>
<td>0.03 ± 0.19</td>
<td>-8.53 ± 0.92</td>
<td>0.36 ± 0.06</td>
</tr>
<tr>
<td>Sparse</td>
<td>Hamm</td>
<td>0.990</td>
<td>0.04 ± 0.08</td>
<td>-9.41 ± 0.49</td>
<td>0.35 ± 0.03</td>
</tr>
<tr>
<td>OVO</td>
<td>Hamm</td>
<td>0.976</td>
<td>0.01 ± 0.08</td>
<td>-10.61 ± 0.91</td>
<td>0.38 ± 0.03</td>
</tr>
<tr>
<td>OVA</td>
<td>Hamm</td>
<td>0.986</td>
<td>1.21 ± 0.33</td>
<td>-11.09 ± 0.78</td>
<td>0.45 ± 0.07</td>
</tr>
<tr>
<td>Dense</td>
<td>Loss</td>
<td>0.990</td>
<td>-0.22 ± 0.08</td>
<td>-19.13 ± 0.96</td>
<td>0.41 ± 0.04</td>
</tr>
<tr>
<td>Sparse</td>
<td>Loss</td>
<td>0.993</td>
<td>0.01 ± 0.10</td>
<td>-17.29 ± 0.78</td>
<td>0.45 ± 0.03</td>
</tr>
<tr>
<td>OVO</td>
<td>Loss</td>
<td>0.980</td>
<td>0.02 ± 0.28</td>
<td>-46.02 ± 3.31</td>
<td>0.50 ± 0.05</td>
</tr>
<tr>
<td>OVA</td>
<td>Loss</td>
<td>0.881</td>
<td>0.04 ± 0.12</td>
<td>1.44 ± 0.21</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 7.7: Accuracy difference depending on the training size ratio $r$ with respect to $r = 0.7$ for four combiner schemes and two decoding schemes. The four left-side columns show the adjusted R-Square showing the goodness-of-fit (1 is best), the minimal difference (MinDiff), the maximal difference (MaxDiff) and the ratio $r_{opt}$ for which the difference is only $y = -0.01$, i.e. only one percent point.
Chapter 7 Results

Summary  In summary, this section showed that if hyperparameter search is carried out, the ensemble strategy does matter, which opposes findings of [15]. Moreover, this thesis could not substantiate the superiority of either Loss or Hamming decoding claimed by Allwein et al. [209]. This once more illustrates the importance of well-tuned binary SVMs.

7.2 Massive feature selection works

After setting up a proper way of performing multi-class support vector machines, the features used within ECG-classification have to be determined. According to chapter four, one can choose between 12 different effect-size indices generating interclass scoring tables and four strategies to derive feature sets from those tables. This section is split into three parts. First the basic experiments used herein are described. Toy Monte Carlo simulations have been conducted for gaining insights into the basic properties of the proposed indices. These properties comprise computational complexity, prediction of SVM output, inter-correlations between those indices and so forth. The results of this analysis are presented in the second part of this section. In the last part, the performance of the different feature selection strategies is evaluated using one-tenth of the feature database consisting of 1.4 million features.

7.2.1 Setup of basic experiments

In this section, two major types of experiments have been conducted: Toy Monte Carlo simulation and ECG verification. The Toy Monte Carlo is intended to assess basic properties of the effect-size indices used within this thesis. The knowledge gained in these two-class experiments is used to understand the results of the verification with real ECG data.

Toy Monte-Carlo simulation  The Toy Monte Carlo simulations are intended to understand basic characteristics of the effect-size indices. This allows for the judging of the performance of the effect-size indices on a neutral basis and favoring certain effect-size indices not only according to the outcome of experiments. Four major questions are addressed by the setup described here:

- Are effect-size indices stable, i.e. the index does not alter dramatically when facing two samples sampled from the same distribution?
- Can effect-size measures predict support vector machine performance?
- How intense is the computation of these indices?
- To what extent do indices correlate with one another?

The basic idea of the Toy Monte Carlo simulation is to generate synthetic features by sampling two series from slightly different configurations of the same probability density function. Afterwards, the power of the feature to separate series one from series two is assessed via the different effect-size indices.

The simulation employs four different basic distributions: the normal, the exponential

\[ P(x) = \frac{1}{\beta} \exp(-\frac{x}{\beta}) \]

the uniform and the logistic distribution

\[ P(x, \mu, s) = \frac{e^{-|x-\mu|/s}}{1+e^{-|x-\mu|/s}} \]

From each of those distributions, two series, \( y_1 \) and \( y_2 \), of size \( 2 \cdot N \) are sampled. The first \( N \) samples are used in the training stage, whereas
the remaining samples are used for testing. Each series is sampled from a distribution with a slightly different configuration\(^6\). The simulation is carried out 1000 times for all four distributions which yields 4000 setups (features) of two series, each with \(N\) samples in the training and the test set.

In the training and testing stage, the discriminatory power of each feature is assessed by effect-size indices \(\text{EFF}_{\text{train}}(y_{1}^{\text{train}}, y_{2}^{\text{train}})\) and \(\text{EFF}_{\text{train}}(y_{1}^{\text{test}}, y_{2}^{\text{test}})\). The stability of a certain index is measured by computing the Spearman rank correlation\(^7\) \(r_{\text{Spear}}(\text{EFF}_{\text{train}}, \text{EFF}_{\text{test}})\). A properly stable index should not change tremendously when facing two samples originating from the same setup. This is quite important. Otherwise, the effect-size indices gained during the feature selection with respect to the training data can be misleading.

The training stage is also used to train a one-feature-SVM, i.e. an SVM using the one-dimensional data (one feature) in the training set to discriminate between \(y_{1}^{\text{train}}\) and \(y_{2}^{\text{train}}\). The mean-per-class sensitivity \(T\)\(^8\) (or balance rate) of the resulting SVM is calculated. A high correlation \(r_{\text{Spear}}(\text{EFF}_{\text{train}}, \text{SVM}_{\text{test}}^{T})\) will indicate that the effect-size index can be exploited to predict SVM outcome. This would be a strong hint that features selected via these indices will turn out to be valuable in the final feature set, which is classified using a SVM.

The last assessment is whether or not there are strong inter-index correlations, i.e. two effect-size measures \(\text{EFF}_i\) and \(\text{EFF}_j\) rate the same features similarly. In that case, they are strongly correlated and thus, \(r_{\text{Spear}}(\text{EFF}_{\text{train}}^{i}, \text{EFF}_{\text{train}}^{j})\) will be high. During all computations, the time necessary to compute the effect-size indices and to train the SVM are calculated and are used to rate their computational intensity.

Optimal setup search The major aim of this section is to show the power of feature selection methods developed in this thesis. The evaluation is based on the feature database calculated from the MIT-BIH Arrhythmia database. It consists of 1.4 million features calculated for around 100,000 ECG-beats. Clearly, selecting the best feature selection strategy based on the full database is likely to cause overfitting. Hence, only a portion (one-tenth) of the database is used. The performance is assessed via the \(\kappa\)-index of a multi-class SVM over ten randomized training-test splits.\(^9\) The SVM uses the OVO scheme with Hamming decoding and the fast histogram hyper-parameter search (Gauss-kernel, MinMax normalization).

The effect-sizes are calculated with respect to the heartbeats used for training of the SVM. Moreover, each index for each feature is a 5-fold cross validated estimate of the effect-size index, i.e. averaging the effect-size over five disjoint sets of heartbeats in the training set. A complete analysis over all 12 indices and all 4 strategies is quite cumbersome. As each of those indices can be parameterized by two or three continuous parameters, the optimal feature selection strategy is a high dimensional optimization problem. This provides lots of opportunities for over-fitting. Hence, the best effect-size index will be selected first by means of the top per class (TopPC) and top mean (TopMean) strategy. This index is used for further evaluation of all four feature selection strategies.

\(^6\) e.g. \(y_1\) is sampled from the normal distribution \(\mathcal{N}(0, 0.5)\), whereas \(y_2\) is sampled from \(\mathcal{N}(\mu, \sigma)\); with \((\mu, \sigma)\) chosen randomly.

\(^7\) The Spearman rank correlation is employed, because the value of the effect-size index is not as important as the rank of each feature. A high Spearman-rank-correlation between two indices indicates that a feature that is rated lower than another feature by the first index is also ranked lower by the other index.

\(^8\) see confusion matrix performance measures in section 2.3.

\(^9\) A major problem in the MIT-BIH Arrhythmia database is that one is faced to make a decision between two suboptimal splitting strategies: Either, one can use only a single training-test split using distinct datasets (supervisedly selected) for only 4 types of Arrhythmias or one can use truly random training-test splits with overlapping datasets but disjoint heartbeats of 13 types of Arrhythmias. A more elaborate discussion can be found in section 2.4 which compares different selection strategies in the field of ECG classification.
Chapter 7 Results

Final feature set determination After the best setup has been determined, the resulting feature selection strategy is applied to the whole feature database using the same test and training splits as before. All those evaluations will be performed using SVMs employing OVO with Hamming decoding based on histogram hyper-parameter search and MinMax normalization. The final feature set is the set with the highest performance \((j_k\)-index) in the ten training-testing splits. The performance of this feature set is evaluated using ten new training and test-splits calculated using Dense ECOCs with Hamming decoding based on histogram hyper-parameter selection using a wider search frame. The final feature set is used to get a cross validation estimate using a completely different ECG database: the MIT Longterm ST database. One should recall that the latter database uses completely different ECG leads. Thus, the cross-validation accuracies should be significantly lower than for the MIT-BIH Arrhythmia dataset. One should also take into account that the second ten-fold randomization is performed on the same database. Hence, the average \(j_k\)-Index over all ten feature sets is the most valid \(j_k\)-estimate. The second ten-fold randomization of the best feature set accounts for the necessity to compare results with other ECG-classifiers. For this purpose, the \(j_k\)-Index of a single feature set is needed. Nonetheless, the difference between both \(j_k\)-Index estimates is negligible. \((93.04 \pm 0.10\%)^{10} \text{ vs. } 92.21 \pm 0.21^{11}\)

7.2.2 Effect-size indices: Basics

The understanding of basic properties of the effect-size indices is vital for the subsequent analysis. Therefore, three properties have been determined: effect-size index stability, prediction power and computational complexity.

Effect-size index stability Effect-size stability is essential for feature set stability and performance. In the training stage, the feature selector uses effect-size indices to build a suiting feature subset. This subset will only render successful if the indices used to generate this set are reliable. The estimation of the indices’ stability was performed using the Toy Monte Carlo mentioned above. For each of the 1000 simulations, 12 effect-size indices have been calculated. The criteria of stability is how strongly effect-size measures calculated in the training set correlate with their corresponding values in the test set. The graph in fig. 7.4 depicts the relation between the Spearman rank correlation, measured between effect-size values in training and testing, and the number of samples in the training set. It demonstrates that effect-size indices based on the assumption of normally distributed data are mainly unstable. It reveals that effect-size indices derived from 2x2 contingency tables, the dichotomized effect size indices (DESI), are much more stable. All indices reach a correlation coefficient of 0.95 or bigger given a training set size of at least 200 samples. This is even superior compared to the stability of the SVM’s balance rate. One needs 800 samples or more to gain a correlation coefficient this high. The standard effect-size indices (except AUC and CD), on the other hand, do not exceed \(r = 0.85\) even at high sample sizes \((N=1000)\). The stability of indices based on correlation coefficients (Pearson, Spearman) is even worse \((r = 0.8)\). With regard to the amount of samples, one can conclude that even at small sample sizes standard effect-size measures are inferior.

The major result of the stability analysis is that DESI are much more stable than standard effect-size indices. And even better: They are more stable and exhibit a stronger stability when facing less data compared to the balance rate of the one-feature SVM.

Predicting SVM outcome Besides stability, one would like to know which of those effect-size indices is promising by means of performance. This begs the question how performance of effect-size

---

\(^{10}\) final feature set derived from static wavelet feature subset

\(^{11}\) one-tenth of database
7.2 Massive feature selection works

indices should be measured without using an explicit feature selection method. Recently, Chang and Lin [181] demonstrated that the weights of a linear SVM are a great predictor of goodness of features. This embedded method showed superior performance over Fisher score and accuracy change by feature removal.[181] In essence, the SVM weights contain feature selection information. As it is not feasible to calculate a linear SVM using over one million features, this road is a dead end. The last paragraph showed a potential way to extract a surrogate of those weights in a computationally cheap way: the one-feature SVM.

The one-feature SVM does not rate the importance of a feature compared to others, but it does assess the performance achievable when solely this feature is used. Now, one can measure the goodness of the effect-size index by its ability to predict the SVM’s outcome. Hence, the Spearman rank correlation between the value of the effect-size index in the training set and the balance rate of the one-feature SVM in the test set is the performance criterion. In easy terms: high effect-size index values in the training set should indicate high performance of the one-feature SVM in the test set.

The results of the Toy Monte Carlo simulation carried out for the stability test can now be reused. One simply measures the correlation between effect-size indices in the training and SVM balance rate in the test set, not between one another. The final correlations are depicted in fig. 7.5. It reveals that DESI measures are superior to standard ones. Their correlation coefficient ranges from 0.4 to a maximum of 0.7. This is pretty low when compared to the 0.82 to 0.88 DESI exhibit. This is a strong hint for the overall superiority of DESI, which will be substantiated later on. This is not so surprising as standard effect-size measures exhibit low stability, whereas SVMs’ balance rate is quite stable. Therefore, a high correlation was not likely. Anyhow let us focus on a much more interesting feature of the graphs in fig. 7.5. One can easily judge by the graphs that, for small sample sizes, DESI do predict SVM outcome even better than the SVM itself. Figure 7.6 presents this relation much clearer. There, the difference
Figure 7.5: Prediction power of effect-size functions: The Spearman rank correlation between effect-size measures calculated in the training set of a two-class problem and the balance rate of an SVM (trained on the corresponding training set) in the test set. As the size of the training set N increases, correlation rises accordingly. The correlation of the balance rate of the SVM in the training set (black) is slightly better than the one with effect-size measures ODD, RSK, F1, ACC, BIN and INF (blue). The correlation in the other six effect-size indices (red) is less pronounced and saturates early. See effect-size definitions on p. 88 and p. 94.
between the correlation coefficient of the one-feature SVM and the DESI functions is depicted. Plainly spoken: the figure below shows the blue lines minus the black line in fig. 7.5.

This experiment reveals that up to \( N = 153 \) samples, the balance rate of the one-feature SVM in the test set is an inferior predictor of the balance rate in the test set. The prediction power of the DESI increases rapidly at lower sample sizes. The correlation can be up to 0.15 points higher than the balance rate's correlation. This is a quite promising result because with regard to the ECG data, one is faced with a particularly small amount of data. As will be shown later on, one needs a reliable benchmark of the goodness of a certain feature discriminating between two classes with few instances.

![Effect size indices predict SVM outcome better than SVM at small sample sizes](image)

Figure 7.6: Difference between the Spearman rank correlation coefficient of the one-feature SVM and the coefficient of ODD, RSK, F1, ACC, BIN and INF. See effect-size definitions on p. 88 and p. 94.

**Computational complexity**  Performance and stability are essential for a good filter method of feature selection. Nevertheless, computation time should be limited. During the Toy Monte Carlo simulation mentioned in the last paragraphs, time measurements have been conducted. The analysis of the timings of the effect-size calculation revealed that there is a linear relation between computation time and size of training set. Thus, a linear fit was applied to the timings. The resulting slope is the increase in computation time per sample and the intercept can be regarded as the minimum or baseline computation time. Both the increase and the baseline values for each index are depicted in the fig. 7.7.

As expected, the SVM is 3 magnitudes slower with regard to minimal computation time and its increase in computation time is 5 magnitudes higher compared to odds, risk or F1 power effect. The comparison is slightly complicated because the computation time of SVMs does increase quadratically. It follows

\[
T(N) = (\text{Increase} \cdot N + \text{Minimum})^2.
\]

**Correlations**  In the last paragraphs, the computational complexity, stability and SVM outcome correlation of twelve effect-size indices have been analyzed. This final paragraph raises the question if any
of the twelve effect size indices shows redundant information. In order to analyze a possible connection between two effect size indices, the Spearman rank correlation between those indices was calculated, employing all Toy Monte Carlo simulations. With regard to tab. 7.8 Hedges G, Cohens D and the Fisher score give redundant information. Moreover, calculating the AUC is as good as calculating the accuracy effect.

7.2.3 Effect-size index performance

Now the main question is how will the effect-size indices perform on real data. The test setup is pretty simple: First the amount of features in the feature database is reduced to ten percent (100.000 features). This is intended to guard against over-fitting. Then, ten test and training splits are generated, which stick to recommendations of [17] (approx. 30-70 split, see the last paragraph in subsection 2.2.1). Each of these splits is henceforth used for feature selection and subsequent performance assessment via a multi-class SVM. The SVM is trained on the same feature vectors that have been used for the feature selection. It utilizes the fast histogram hyper-parameter selection with two-fold cross validation, MinMax normalization and the OVO multi-class scheme with Hamming decoding.

The feature selection process is divided into two parts: generation of inter-class scoring tables and subsequent evaluation using three different strategies: top per class, top mean and table filling. A major concern during feature selection is the stability of the feature sets of the 10 train-test splits, i.e. the feature sets should share a great portion of features. This is taken into account by employing cross validation of the effect-size indices. Therefore, each effect-size index is calculated on 5 subsets of the training set. The final score is the average over those subsets.
### 7.2 Massive feature selection works

<table>
<thead>
<tr>
<th></th>
<th>HG</th>
<th>HGU</th>
<th>CDP</th>
<th>FS</th>
<th>CD</th>
<th>ODD</th>
<th>RSK</th>
<th>F1</th>
<th>ACC</th>
<th>AUC</th>
<th>BIN</th>
<th>INF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HG</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.86</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>HGU</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.86</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>CDP</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.86</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>FS</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.86</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>CD</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.86</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>ODD</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.95</td>
<td>0.87</td>
<td>1.00</td>
<td>0.97</td>
</tr>
<tr>
<td>RSK</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>1.00</td>
<td>0.00</td>
<td>0.95</td>
<td>0.95</td>
<td>0.87</td>
<td>1.00</td>
<td>0.97</td>
</tr>
<tr>
<td>F1</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.95</td>
<td>0.95</td>
<td>0.00</td>
<td>1.00</td>
<td>0.89</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>ACC</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.95</td>
<td>0.95</td>
<td>1.00</td>
<td>0.99</td>
<td>0.90</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>AUC</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.87</td>
<td>0.87</td>
<td>0.89</td>
<td>0.90</td>
<td>0.00</td>
<td>0.87</td>
<td>0.90</td>
</tr>
<tr>
<td>BIN</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.95</td>
<td>0.87</td>
<td>0.00</td>
<td>0.97</td>
</tr>
<tr>
<td>INF</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.97</td>
<td>0.97</td>
<td>1.00</td>
<td>0.99</td>
<td>0.90</td>
<td>0.97</td>
<td>0.00</td>
</tr>
<tr>
<td>∅</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 7.8: Spearman rank correlation between the values of effect-size indices in the Toy Monte Carlo experiments. See effect-size definitions on p. 88 and p. 94

#### Accuracy and stability follow exponential resp. linear growth functions

The evaluation of the best feature selection strategy necessitates computing three different feature selection strategies with regard to inter-class scoring tables of 12 effect-size indices. Additionally, this has to be carried out using several setups of those strategies. For each setup the $\kappa$-index (weighted accuracy) and the Average Tanimoto Index (feature set stability) are evaluated together with the mean number of features used over all 10 train-test splits.

Hence, feature selection strategies are judged by $\kappa(F_s)$ and $ATI(F_s)$; the dependence of accuracy and stability on the number of features selected $F_s$. An optimal method would have a rapidly increasing $\kappa(F_s)$ which saturates early at a high level. Moreover, the underlying 10 feature sets should show a decent similarity, which should increase while feature set sizes rise. In this scenario (increasing ATI), the feature selection chooses from a finite pool of features which are high-performing throughout all splits. For a better understanding, fig. 7.8 depicts the $\kappa(F_s)$ and $ATI(F_s)$ functions for the TopPC strategy for chosen effect-size indices.

With regard to fig. 7.8, $\kappa(F_s)$ follows an exponential growth function, which was previously used in the SVM case (see eq. 7.2). Moreover, it demonstrates that the "parametric" effect-size functions: Fisher-Score, HedgesG and CohensD are inferior compared to the dichotomized effect-size indices (DESI). One should have ascertained this by simply looking at the results of the Toy Monte Carlo experiments. There, the DESI parameters showed great capabilities to predict the SVM outcome. Hence, one can conclude that a high correlation with SVM outcome on a per feature basis is a strong predictor for performance in the multi-class scenario. Unfortunately, high performance is not the only concern. The feature set stability measured by $ATI(F_s)$ is higher in "parametric" effect-size indices.

#### A quantitative comparison

The latter remarks are purely qualitative. Hence, a quantification is necessary, which allows one to easily visualize the trade-off between high feature set stability (high ATI) and high performance (high $\kappa$). It is of great advantage that both $ATI(F_s)$ and $\kappa(F_s)$ follow clearly defined functions: linear resp. exponential growth. Hence, the fitting parameters can be used to assess the asymptotic $\kappa$-index and the asymptotic maximal number of features.

The maximal $\kappa$-index is clearly identified with $y_0$ in eq. 7.2, denoting the upper bound of the exponential growth. So what is the maximum number of features used. One can see that the ATI graphs are constantly growing linearly. Let us keep in mind that the ATI is defined as the average similarity between two feature
Figure 7.8: The \( \kappa \)-index (top) and average Tanimoto index (bottom) in dependence of the amount of features selected. \( \kappa(F_s) \) is regressed by an exponential growth function, whereas \( ATI(F_s) \) uses a linear regression. The top plot consists of all 12 effect-size indices, whereas the bottom plot shows only three effect-size indices, which comprise the possible values of \( ATI \).
sets, resulting from one of the ten splits with one another. A perfectly stable feature selector would have ATI equal to one. As the graphs are increasing towards one, one can reason that the features are selected from a finite pool of around 500 features. The size of this finite pool is what I call the asymptotic number of maximal features. This number can be calculated by extrapolating the linear fit towards $ATI = 1$:

$$ATI(F) = m \cdot F + n \rightarrow ATI(F_{\text{max}}) = 1 \rightarrow F_{\text{max}} = \frac{1 - n}{m}$$

(7.5)

Obviously some of the graphs do fit better to the linear model. Thus, the slope ($m$) and intercept ($n$) can be determined more easily. This should be taken into consideration too. According to Gauss rule of error propagation, the error on $F_{\text{max}}$ is:

$$\Delta F_{\text{max}} = \sqrt{\left(\frac{\partial F_{\text{max}}(m, n)}{\partial m} \Delta m\right)^2 + \left(\frac{\partial F_{\text{max}}(m, n)}{\partial n} \Delta n\right)^2}$$

(7.6)

$$= \sqrt{\left(\frac{n - 1}{m^2} \Delta m\right)^2 + \left(\frac{-\Delta n}{m}\right)^2}$$

(7.7)

One inserts the errors from the linear fit $\Delta m$ and $\Delta n$ and gets an estimate of $\Delta F_{\text{max}}$. Figure 7.9 below shows the juxtaposition of $F_{\text{max}}$ and $y_0$ for the TopPC and TopMean strategy. I refrain from showing the TopOOO results, as they are not as strictly linearly resp. exponentially growing with regard to $ATI$ resp. $\kappa$.

The first noteworthy aspect of the graphs is the superiority of TopPC vs. TopMean. Even though the TopPC strategy yields less stable feature sets (black box slightly shifted to the right), the $\kappa$-index is around two percent points higher. Another striking feature is that the asymptotic performance of the effect-size measures from dichotomized continuous variables show higher $\kappa$-index than those from parametric effect-size indices. The Fisher-Score shows low performance in both schemes. Let us look at the detailed analysis in the picture below. It shows that the two best features are clearly InformationEffect and RiskEffect. The latter also shows high performance in the TopMean scheme. The AccuracyEffect will not be mentioned further as it yields very instable feature sets. (high error on $F_{\text{max}}$)

In the Toy Monte Carlo section, it was demonstrated that DESI are powerful effect-size measures. They provide high stability and a potential prediction power for SVM outcome. Their computational complexity is low enough to be calculated in considerable time. Now, one can add that they are also superior in real-life applications using thousands of features.

### 7.2.4 Comparison of feature selection strategies

After finally opting for Risk Effect as the effect-size index of choice, one can compare the four proposed feature selection strategies with common methods of feature selection (FS). This experiment, still, employs 100,000 features only. The four FS strategies are compared to a standard feature-filtering technique: the Fisher-discriminancy described in chapter five. The first striking feature of the upper panel of fig. 7.10 is that the FisherLDA strategy is by all means inferior compared to the other four strategies. Not only is its $j_k$-Index up to thirty percent points lower than the TopPC strategy, its Average Tanimoto Index (see fig. 7.11) drops with increasing number of selected features. This means that the FisherLDA
Chapter 7 Results

Figure 7.9: Comparison of twelve effect-size measures for two feature selection techniques: the TopMean (red) and TopPC (strategy). The plot above shows the Asymptotic Kappa Index ($\kappa_{\text{max}}$) with respect to the asymptotic number of features ($F_{\text{max}}$).
does not produce stable feature sets. Moreover, the selected sets show poor classification performance. Maybe, this is only an effect of the Fisher score being an inferior effect-size measure. As one can deduce from Toy Monte Carlo simulations, as well as from ECG data experiments, that the Fisher score does not perform well (see fig. 7.9).

**FisherLDA is inferior**  Hence, a comparison of the FisherLDA and other feature selection techniques based on the Fisher score should show low performance for all methods. Indeed, the Fisher score processed via the TopPC and TopMean strategy performs as bad as the FisherLDA (see fig. 7.12). The only difference is that the FisherLDA shows a much higher standard deviation and has a slightly smaller performance at smaller feature set sizes. This lack of classification power is compensated by larger feature sets. The major reason for the high variability of the $\kappa$-Index is the low mean-per-class sensitivity of the FisherLDA, even though the accuracy of the FisherLDA reaches up to 95%. This problem can be easily understood from the definitions of the multi-class Fisher discriminancy, which is the basis of the FisherLDA. The FisherLDA does not sum up or average over effect-size measures of a single class. It simply sums up the inter-class and intra-class scatters. In case of a highly skewed distribution of the amount of samples per class, the class with the highest number of instances potentially dominates the fisher discriminability. The TopPC strategy on the other hand focuses specifically on summing up the effect-sizes on a per class basis. In fact, the mean sensitivity and accuracy are not altered by this approach due to the weak performance of the Fisher discriminancy. The variability of the sensitivity and accuracy, on the other hand, are heavily reduced because the TopPC strategy does search for features that are capable of more than simply separating the dominating normal-beats from the other ones.

Now, let us take a closer look at the lower panel of fig. 7.10. It shows the $\kappa$-Index of the four feature selection techniques developed in this thesis using the RiskEffect effect-size. One easily verifies that the TopMean strategy has a lower performance than TopPC, TableFill and TopPC with correlations strategy. This is true for all generated feature set sizes. The TopPC and TableFill strategy have comparable performance, whereas the additional consideration of correlations between features increases performance further. The $\kappa$-Index is only one side of the problem. The other one is the Average Tanimoto Index (ATI) depicted in fig. 7.11. It shows that the high performance of the TopPC + Corr strategy is traded for a lower stability of the feature sets. The graph also shows the increased instability of the size of the feature sets in the TableFill strategy. It shows the highest variability compared to all other four selection schemes.

In conclusion, the ATI of the TopPC+Corr strategy remains high and is slightly lower than for the TopPC technique. On the other hand, the performance of the $\kappa$-Index is increased by up to 0.5 percent points. Therefore, the TopPC+Corr strategy is used for the experiments to come.

### 7.3 Performance of Feature Extractors and normalizations

In the last section, I could show that the TopPCCorr strategy is superior to the other three developed selection schemes and is clearly superior over the feature selection proposed by Yu and Chen [118]. With this powerful tool at hand, one can assess the performance of the feature-extractors mentioned in chapter 4. The main findings of my analysis are as follows:

- RR-Interval parameters are complementary only to features solely focusing on QRS-shape.  

---

12 This is according to Inan et al. [36] who presented a tremendous increase in accuracy due to using RR-interval information.
Figure 7.10: Overview of four feature selection methods developed in this thesis and standard feature selection using Fisher-Discriminancy with regard to the $\kappa$-index: The $\kappa$-index depending on the number of features each method provides. In the top panel, the TopPC + Corr (black), TopPC (orange) and FisherLDA (red) are fitted via a logarithmic function and the TableFill and TopMean methods via an exponentially decaying one. The bottom panel zooms into the high performance region (90-94%). There, the $\kappa$-index is fitted using the logarithmic function. The first point was masked for TopMean. The first two points were masked for TableFill.
7.3 Performance of Feature Extractors and normalizations

Figure 7.11: Comparison of four feature selection methods developed in this thesis and a standard feature selection using Fisher discriminancy with regard to the Average Tanimoto Index (ATI): the TopPC, TopMean, TableFill, FisherLDA and TopPC strategy with correlations (orange, olive, blue, red, black).

Figure 7.12: Comparison of three feature selection techniques based on the Fisher-Score with regard to the $j\kappa$-index: the TopPC, TopMean and FisherLDA strategy.
Physiologic delineation is superseded by static delineation.

- The normalization of the ECG of a single heartbeat plays a minor role.

- The third-order spectra contains useful information

- Features based on à trous - wavelet decomposition are superior to ones based on discrete wavelet decomposition.

- Different static features need different preceding FIR filters.

- Auto-Regressive Models show inferior classification performance.

**Experimental setup**  Before I start with some graphs supporting those claims, let us recapitulate the experimental setup. In essence, the feature-extractors have been grouped into 19 sub-groups according to the feature set database. The TopPC+Corr algorithm was performed to select the best features within each group. The final configuration was \( \text{TopPCCorr}(N_{\text{top}} = 4, N_t = 400, r_{\text{max}} = 0.5) \) which is exactly the setup used for generating the curves in the feature selection experiments. Again, the same 10 training and test-splits have been used.

**RR-Parameter performance**  The first thing I analyzed was whether or not adding RR-interval parameters to the feature groups actually does increase their performance. RR-interval parameters are of particular importance because they presumably contain extra information. The majority of the feature-extractors assess the shape of the QRS-morphology without regard to nearby ECG-beats. Therefore, contextual information is missing. The figure below shows the impact of adding the RR-intervals.
7.3 Performance of Feature Extractors and normalizations

The plot shows that for nearly all feature-extractors, adding RR-interval features reduces the feature set size and increases classification performance. It is also astonishing as well as reassuring that even though RR-interval parameters account for only 0.002% of the total feature set, the feature selection scheme extracts exactly this necessary information. One can see that auto-regressive models and the raw physiologic information (PQRST amplitudes and timings) benefit greatly from RR-interval parameters. This also holds for morphologic features considering the width or asymmetry of the QRS-complex only (Asy and Wi extractors). The performance ($\kappa$-index) of each of those feature-extractors can be increased by at least 2 percent points. This supports the hypothesis that the RR-interval features are complementary to morphologic features.

The importance of features assessing the area in between characteristic or static points of the ECG can also be stressed with regard to the above figure. The main information does not reside at some special point. The area between those points is crucial for the understanding of the time-course. Thus, the PhysioArea extractor, which calculates the partition functions on several areas of physiologic interest, takes only a slight advantage of the RR parameters.

**Static delineation is more powerful than physiologic**  The static delineation parameters as well as those derived from third-order spectra and wavelet decomposition do not benefit a great deal. This has several reasons. Topmost, all those feature-extractors use fixed segment sizes. A variety of arrhythmias is characterized by the notion of prematureity. This means that an ECG-beat departs from its typical behavior, i.e. the ECG-beat occurs much earlier than to be expected. In the case of physiologic delineation, the segment of interest scales with the width of the QRS-complex and with respect to a potential prematureity. This is due to the fact that physiologic delineation considers the PQRST points of the very ECG-beat it

Figure 7.14: The change in $\kappa$-Index due to an additional use of RR-Interval parameters. Description of used feature extractors can be found in section 4.8. All extractors have been applied to all available segments.
investigates on. No information about the distance of the P-wave to a previous T-wave or QRS-complex can be extracted that way. Fixing the segment size allows exactly those situations to be tracked down.

Three different segments have been chosen: one with a duration of 833 ms around the R-peak, one with 416 ms around it and one with 500 ms in between heartbeats. The choice of this segment ensures contextualization of the information gained by the feature-extractors. A premature excitation of the myocardium can be easily tracked, as it will show up in one of the longer or in the between beats segment. This is also an explanation why the additional RR-interval information does not increase the classification performance tremendously.

Stationary vs. discrete wavelet decomposition

In this thesis, great efforts have been made to analyse the time-frequency behavior of the ECG. The usage of the short time Fourier transform was ruled out early by theoretical considerations (see chapter 4). Hence, the wavelet transform is the main tool for this investigation. Therein, a differentiation between two main branches, the à trous and the Mallat scheme of wavelet decomposition is necessary. The latter leads to the popular discrete wavelet decomposition (DWT), whereas the other scheme conducts the so-called stationary wavelet transform (SWT). The DWT splits the signal into segments of increasing time and decreasing frequency resolution and is not time-shift invariant. The SWT on the other hand is time-shift invariant, i.e. the position of the R-peak does not change in transformed signal. Moreover, the segment sizes are totally different. A four-staged DWT yields 5 segments. If one combines those segments, they are as long as the ECG segment itself. The SWT on the other hand returns 5 segments, each of the same length as the ECG segment. This is to be expected, as the DWT performs a change of basis of the ECG, whereas the SWT is a representation of the ECG using a linear hull.

The major advantage of using the SWT is that one can apply the same feature extraction techniques to the transformed signal that have been employed for the ECG itself. The transformed signal is basically the output of successive high-pass filters applied to the ECG. Now, what kind of transform is more powerful? The plot below is pretty clear about that. Features derived from the DWT (Wavelet) show a lower $\kappa$-index and produce slightly larger feature sets than those derived from SWT(Dyadic). The power of the SWT becomes more striking if one compares the parameters derived from static delineation (Morphostatic) and those based on static delineation subsequent to several stages of the SWT (Dyadicstatic). The graph demonstrates that for many static delineation strategies, a proper filtering of the ECG is available. It is not surprising that each feature-extractor has an optimal frequency band in which it can extract the most information while being shielded against high amounts of noise.

Let us take a look at the morphologic features concerning the shape of the QRS-complex. It seems that applying FIR filtering prior to the calculation of the asymmetric width measures (SWT+Asy, QRS+Wi) is more favorable compared to the calculation of the symmetric width measures (SWT+Wi, QRS+Wi). With regard to the figure above, one clearly sees that, after filtering, the performance of the asymmetric and symmetric width measures (SWT+Asy+Wi) move closer to each other. Supposedly this originates from the fact that the asymmetric width measures are more easy distorted by noise than the symmetric width measures. One can also see that after filtering the information content in the symmetric ones is reduced, which corresponds to a reduced $\kappa$-index. In simple terms, removing noise from the width measures reveals their inferiority compared to the asymmetric ones, which show their potential only if noise is removed properly.

Normalization: There is no free lunch

In the preprocessing chapter, the possibilities of normalizing were discussed and three different normalizations, which employ basic statistics, were proposed. Even though more advanced normalization techniques like morphing could be carried out, this thesis
will focus on those less sophisticated but robust methods. As far as the experiments with those basic normalizations can tell, classification results stay the same whether applying a normalization or not. Why is it like that? The intention of performing normalizations is to reduce inter- and intra-patient variability. The shape of the QRS-complex depends heavily on the electrical axis of the heart, the exact position of the electrodes and the quality of the contact between the electrodes and the skin. Therefore, normalization based on single lead will inevitably lead nowhere. In a next step, different possibilities for multi-lead normalizations and adjustments should be carried out. This large scale analysis shows that the normalizations do have an impact on the features. Otherwise the same features would be extracted, which they are not. However, for each normalization, features can be found which render equally capable of classifying the heartbeats. There is no free lunch. The proposed normalizations will decrease inter- and intra-patient variability for some patients and some arrhythmias, but will increase variability for others. Simple normalizations that do not account for the wider context of the ECG measurement will not increase the performance.

**Cumulants and Auto-Regressive models** One main question in this thesis was whether or not the third-order spectra contains viable information. This thesis expanded the possibilities of feature extraction of the third-order spectra by using different segments of interest based on a tiny portion of the available data. In the overview given above, the features extracted from cumulants show a performance which is lower than the one for wavelets, but still has greater classification capabilities than normal physiologic delineation strategies. Thus, in contrast to auto-regressive models, it can be considered a powerful source of information. The latter shows very low performance and should not be investigated further.
7.4 The final feature set

In the last section, it has been demonstrated that features solely based on partition functions applied on partitions derived from static wavelet transformed (SWT) ECG segments show great performance. Moreover, they are easy to calculate. Thus, the final feature set is based on just these features. The final set was chosen to be the feature set with the highest $\kappa$-Index of all ten training-test splits used for feature selection. This feature set is now evaluated on ten new training-test sets to estimate its capabilities.

This procedure has been carried out for three feature selection configurations: TopPCCorr(1, 0.5), TopPCCorr(2, 0.5) and TopPCCorr(4, 0.5). The latter two feature sets can be found in A.1 and A.2. The first one is shown below:

\[
\begin{align*}
RRRel5 & \\
Y(\Delta_{seg} = 300, MMNorm) \rightarrow SWT(1) \rightarrow Cut_{3} & (7.8) \\
& \rightarrow SWT(3) \rightarrow StdDev_{5} & (7.9) \\
& \rightarrow AbsMean_{8} & (7.10) \\
& \rightarrow SWT(4) \rightarrow TrendCut_{3} & (7.11) \\
Y(\Delta_{seg} = 150, MMNorm) \rightarrow SWT(2) \rightarrow Max_{16} & (7.12) \\
& \rightarrow TrendSlope_{3} & (7.13) \\
& \rightarrow SWT(3) \rightarrow Min_{8} & (7.14) \\
& \rightarrow SWT(4) \rightarrow MiMax_{16} & (7.15) \\
& \rightarrow ArgMiMax_{3} & (7.16) \\
Y(\Delta_{seg} = 300, NONorm) \rightarrow SWT(3) \rightarrow TrendSlope_{3} & (7.17) \\
Y(\Delta_{seg} = 150, NONorm) \rightarrow SWT(4) \rightarrow MaxAbs_{3} & (7.18) \\
Y(\Delta_{seg} = 300, MedNorm) \rightarrow SWT(4) \rightarrow AbsMean_{3} & (7.19)
\end{align*}
\]

The syntax of each feature follows the scheme:

\[
\text{segmentation} \rightarrow \text{transformation} \rightarrow \text{function}_{\text{partition}} \quad (7.21)
\]

The segmentation stage is carried out by one of seven segmentation approaches explained in the summary of section 3.4. The static wavelet transform (SWT) and its four resulting sub-bands are discussed in section 4.4.5 and on page 81. The notation of the partitioning-section is straightforward: $Min_{2}^{5}$ denotes the minimum in the 2. partition yielded by splitting the ECG-segment into 3 equally sized partitions. For more information on partitioning and the partitioning-functions used below see section 4.2.

**Features are physiologically sound** One major concern during the thesis was if the feature selection would yield abstract features or ones that can be physiologically attributed. With regard to the above thirteen features, the latter is quite true. The RR-Interval relative to the average of the last 5 heartbeats (RRRel5) is considered a strong predictor for prematurity of heart beats. The minimum of the fourth segment in a split of eight ($Min_{8}^{4}$) is connected to the drop of the Q-point. $MiMax_{16}^{6}$ rates the height of the
7.4 The final feature set

<table>
<thead>
<tr>
<th>Stage of SWT</th>
<th>3-dB bandwidth in Hz</th>
<th>Features contributing to feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWT(4)</td>
<td>6 - 19</td>
<td>TopPCCorr($N_{top}$, $r = 0.5$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$N_{top} = 1$</td>
</tr>
<tr>
<td>SWT(3)</td>
<td>12 - 39</td>
<td>5/12</td>
</tr>
<tr>
<td>SWT(2)</td>
<td>26 - 84</td>
<td>4/12</td>
</tr>
<tr>
<td>SWT(1)</td>
<td>90 - 180</td>
<td>2/12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1/12</td>
</tr>
</tbody>
</table>

Table 7.9: Amount of features originating from a certain SWT sub-band for the three final feature sets. The second column shows the frequency content of the sub-band.

P-wave and Max$^{9}_{16}$, in combination with AbsMean$^{2}_{23}$, assesses the height of the R-peak. Let us take a look on the SWT sub-bands employed. Table 7.9 reveals that the majority of features (65 to 75%) is derived from the SWT(3) and SWT(4) sub-bands. For the small ($N_{top} = 1$) and the large ($N_{top} = 4$) feature set the SWT(4) sub-band contributes nearly half of the features. This is no surprise, as the SWT(4) sub-band matches the frequency range of the QRS-complex from 2 to 20 Hz nicely. Accordingly, features describing the high-energy content (SWT(1)) play only a minor role. Only eight features make use of the first stage of SWT. Six of them are concerned with the center segment, which presumably contains the QRS-complex and therefore the highest physiologic frequency contribution. In conclusion, one can argue that the features chosen by the feature selection process are physiologically sound. Thus, they reflect properties of the underlying physiologic problem rather than fitting the data at hand.

Feature set performance The performance of the selected features has been calculated by 10 new training and test-splits of the original data. Table 7.10 shows the performance of the feature sets compared to ECG-classifiers proposed by other authors. A more detailed scoring can be found in the appendices. The results reveal that the approach in this thesis is superior to all reviewed publications by means of the $j\kappa$-Index, accuracy and $\kappa$. Only Niwas et al. [71] present a higher mean sensitivity. However, their $\rho$-Score is quite low (10) because the amount of data is very small (less than one percent of MIT-BIH Arrhythmia data) and the training set is even bigger than the test set. This superiority is even accomplished at a higher $\rho$-Score. For more information about the error estimates of the $j\kappa$-Index and other parameters, see the appendices.

The approach by Prasad and Sahambi [17] uses the exact same amount of data and the exact same training-test split. Given that the mean sensitivity is quite close between this classifier and the approach in this thesis, one might argue that the mean positive predictive value of Prasad and Sahambi [17] should be quite low, because its accuracy is significantly lower compared to the classifiers in this thesis.

7.4.1 Cross-Validation with the LTST database

A simple cross validation test was performed using the data in the LTST-database. The performance is shown below. The $j\kappa$-Index is 74.39% and is significantly lower than the 95.02% achieved on the MIT-BIH Arrhythmia database. This huge difference occurs due to several reasons. First, the LTST database does not employ the same ECG-lead as the MIT-BIH Arrhythmia database. That is why the features selected above are only valid when used in combination with the MLII-lead. As explained earlier in the introduction, the MIT-BIH Arrhythmia database has no "sister" database which has comparable

---

13 An earlier proof of concept version of the feature selection approach using the TableFill algorithm was published in 2012 (see Rockstroh et al. [286]). The feature sets provided therein were calculated on the full database and without using QRS-complex detection. Hence, a higher but less realistic performance could be achieved.
Chapter 7 Results

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>$\rho$-Score</th>
<th>$j\kappa$-Index</th>
<th>$C$</th>
<th>$T$</th>
<th>$P$</th>
<th>$\text{acc}$</th>
<th>$\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>This thesis’ classifier when detecting the unknown beats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TopPCCorr(2, 0.5)</td>
<td>23</td>
<td>92.43%</td>
<td>13</td>
<td>86.80%</td>
<td>88.17%</td>
<td>98.88%</td>
<td>97.38%</td>
</tr>
<tr>
<td>TopPCCorr(4, 0.5)</td>
<td>21</td>
<td>93.04%</td>
<td>13</td>
<td>87.39%</td>
<td>89.39%</td>
<td>99.01%</td>
<td>97.69%</td>
</tr>
</tbody>
</table>

| This thesis’ classifier ignoring unknown beats |              |                 |     |     |     |             |         |
| TopPCCorr(2, 0.5)    | 19           | 94.28%          | 12  | 90.97% | 91.17% | 98.93%  | 97.50% |
| TopPCCorr(4, 0.5)    | 17           | 95.02%          | 12  | 91.92% | 92.54% | 99.07%  | 97.81% |

| ECG-classifiers detecting more than 7 classes with $\rho$-Score > 9 |              |                 |     |     |     |             |         |
| Ince et al. [163]    | 24           | 69.40%          | 8   | 60.79% | 62.79% | 93.63%  | 77.01% |
| Lagerholm et al. [47](i) | 22           | –.–%            | 16  | 67.25% | –.–%  | 98.49%  | 97.05% |
| Rodriguez et al. [5] | 18           | 88.82%          | 13  | 86.97% | 79.59% | 96.72%  | 94.37% |
| Lagerholm et al. [47](ii) | 18           | 92.41%          | 13  | 82.30% | 93.08% | 98.54%  | 97.14% |
| Prasad and Sahambi [17] | 17           | –.–%            | 13  | 86.62% | –.–%  | 96.77%  | –.–%  |
| Niwas et al. [71]    | 10           | –.–%            | 10  | 96.06% | –.–%  | 96.46%  | –.–%  |

Table 7.10: Comparing the performance of the reviewed ECG-classifiers with those in this thesis.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>True</th>
<th>$N$</th>
<th>VPC</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1104349</td>
<td>1304</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td>VPC</td>
<td>57</td>
<td>8435</td>
<td>3298</td>
<td></td>
</tr>
<tr>
<td>OTHER</td>
<td>49</td>
<td>2240</td>
<td>1947</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.11: The performance of the selected feature set TopPCCorr(4, 0.5) using a different ECG-lead, from a different ECG-database, with a different sampling both in the time and voltage domain.

amounts of arrhythmias recorded with the same ECG-lead. Second, the group of patients recorded is quiet different. The LTST database contains patients with an elevated ST-segment which might distort the features that are attributed to certain arrhythmias. Third, the prior probabilities are completely different: In the MIT-BIH database VPCs make up for one tenth of the database. In contrast, VPCs are only one hundredth of the LTST database. Fourth, the feature selection focused on a wide range of arrhythmias. Consequently, the features selected for VPC-detection are one tenth of the overall feature set. Hence, the proposed classifier is not the strongest VPC-detector.

7.5 Current ECG-classifiers lack from reliability and performance

As I laid out in chapter two, the key to assessing classification performance is measuring the $j\kappa$-index proposed by Mar et al. [7] and generalized in this thesis. The reliability of those classification results can be measured using the $\rho$-Score developed in this thesis. I scored 72 publications from the last two decades, according to the scheme described in chapter two. The detailed $\rho$-Score ranking and the individual scorings of each ECG-classifier can be found in the annexes. This review will contain mostly publications after 2002, because the MIT-Arrhythmia database had huge impact on the field and was the first database that made it possible to investigate and compare many classifiers’ performance on the same database. Earlier work was be cited in the feature extraction and feature selection section. Some really high-performing classifiers prior to 2002 are included too[47].

This section is intended to give a brief overview about the current state of the art in ECG-morphology
I will not focus on the different feature extraction and preprocessing techniques, but rather on the data, the process of training and test-set creation and the results themselves. Even though a recent review by Salem et al. [288] summarizes the latest progress in the field of ECG-classification, it does not distinguish between papers based on less or more data, neither does it reflect on the amount of features, the composition of the training and test sets or the selection process of the data. Moreover, it compares accuracies that are in particular not equally defined in the papers under investigation. Furthermore, some papers did not even target the same ECG-classification task.

Excluded publications The major goal of this review is to compare arrhythmia classifiers. Hence, some classifiers have been excluded, because they simply perform QRS- or P-wave detection [53], focus on VF, AF, VT rhythm detection [57, 62, 161, 204, 217], ischaemia (myocardial infarction) detection [57, 113, 233, 289] or give no performance measures at all [31, 127, 290]. Other authors try to determine the disease of the patients by looking at only a few heartbeats. [32, 125, 132]. The proposal of Irvine et al. [112] was not assessed as it deals with exploiting ECGs to identify humans.

Results The analysis of the ECG-classification approaches reveals that the $\rho$-Score and the $j\kappa$-index are anti-correlated, i.e. more reliable classifiers perform worse and "perfect" classifiers are highly unreliable. It is common sense to doubt perfect results and ask how they are obtained. But still high performance and reliability are not contradictory. This thesis provides an objectivation, the $\rho$-Score, which rates the most core necessities for judging ECG-classifiers.

The figure below depicts the anti-correlation of the $j\kappa$-index and the $\rho$-score. One can clearly see that the frontier of well developed ECG-classifiers [5, 34, 47, 68, 85, 111, 291] forms the right edge of the broad stripe. Any new ECG-classifier should be right of this line, i.e. be more accurate at a high reliability level or to be more reliable at a high performance level. The plot moreover shows that the feature sets developed in this thesis clearly exceed the current frontier of classifier performance and reliability.

The anti-correlation shown above would be hidden if one had used the accuracy or sensitivity as the $y$-axis. This is because sensitivity and accuracy measures alone can potentially obfuscate low real-life performance. As explained in more detail in chapter one: the accuracy does not take into account how likely it is that this accuracy could be obtained by a "random"-classifier. A HIV-test showing 99.9% accuracy, when faced with 1 HIV patient in a test group of 999 non-HIV patients, is just what you would expect if you knew the composition of the test population. The $\kappa$-Index accounts for that and normalizes the accuracy to the group composition at hand.

Inevitable errors One might wonder why the proposed feature sets are not performing even better. With respect to feature extraction and feature selection nearly all stops have been pulled out. The SVMs are well tuned and according to best practice. Let us step back for a moment, ignoring the accuracy and $\kappa$ numbers, and take a look at the errors itself. Referring to the confusion matrices in tab. B.1, tab. B.2 and tab. B.3, the majority of errors can be attributed to normal-beat misclassifications. Either normal beats are classified as non-normal ones, the false negatives (FN), or non-normal beats are supposed to be normal ones, the false positives (FP). Figure 7.17 depicts a selection of normal beats detected by the final classifier as well as false positives and negatives. One can clearly see that FNs, from an automation perspective, should not have been labeled normal in the first place. There is pretty much

---

\[14\] A less comprehensive meta-study using fewer ECG-classifiers was published here: [287]. This publication refers to the $\rho$-Score as G-Score. This is an earlier choice of name. It was dropped because rating the generalizability (G-Score) is only one aspect of the $\rho$-Score.
Figure 7.16: Overview of the performance of state-of-the-art ECG-classifiers and their respective $\rho$-score.
7.6 The myth of the perfect wavelet

In the last section it has been shown that wavelet parameters are not superior to other features. Moreover, why is there no superior wavelet as proclaimed in literature [70]. From a feature selection perspective, it has been demonstrated, that no wavelet is superior compared to all features derived from the wavelet transform. More so, it has been discovered that features from different wavelets compiled together perform best. Hence, an explanation is needed. In chapter four, wavelet transforms have been extensively described. One major application of WT is data compression, i.e. WT allows the data to be represented by less information bits. In literature, the compression ability of the WT was studied. Froese et al. [70] proposed an optimization approach for wavelet transforms to increase the amount of variance stored in a minimal set of wavelet coefficients.
Chapter 7 Results

If an optimal wavelet for ECG-classification exists and this optimality corresponds to a maximal compression rate of the wavelet transformed data:

- the wavelet accounts for all different time-frequency behaviors of the different arrhythmias
- the compression rate is minimal for all arrhythmias

**Investigated wavelets** The investigation of the aforementioned problems was carried out as follows: First, the signal was preprocessed according to chapter three. Afterwards, the heartbeats were split into 13 groups. Twelve of them account for the 12 arrhythmias analyzed in this thesis without the unknown beats. The last group consists of all beats in the 13 groups combined. The $CR_i$ is calculated for each beat in each group. The major question is: Do certain wavelets (different $\theta$) have superior coding performance compared to others? Thus, the performance of 22 different wavelets:

- db2, db6, sym6 wavelets on two and three stages
- daublets (db1, db2, db3, db4, db6, db11) on four stages
- symlets (haar, sym2, sym3, sym4, sym6, sym11) on four stages
- coiflets (coif1, coif2, coif3, coif4) on four stages

This investigation will not only focus on four staged WTs, even though this is commonly performed for ECG-analysis. The difference between two-, three- and four-staged analysis is one of the questions addressed by the following experiments. Therefore, the most popular wavelets for ECG-analysis are chosen: db2, db6 and sym6 (see chapter 4). The four-staged setups are analyzed using the first four daublets, wavelets and coiflets. Higher order wavelets like db6, db11 and their corresponding symlets are also examined to cover a wide range of wavelets used in literature.

**The test setup** The test setup is carried out to answer the question: Is it possible to determine a wavelet optimal for ECG-classification by means of optimal compression rate. This is argued by several authors topmost Froese et al. [70]. In ECG-classification, one searches for a transformation in which all arrhythmias become easily separable. Clearly, a high compression rate calculated over all beats under investigation will cause a bias towards the dominant type of heartbeat, namely the normal beat. The main question is, if the time-frequency behaviors of the other arrhythmias are similar enough to ensure a high compression rate over all arrhythmias. With regard to ECG-classification, one is moreover interested in a per-beat compression rate not the total compression rate.

The setup therefore calculates the compression rate for each instance of a certain heartbeat type. A wavelet has been considered superior over another wavelet with regard to the compression ratio $\alpha$, if:

- there is a statistically significant difference on a beat-to-beat basis between the wavelets
- for at least 50% of the heartbeats, the compression ratio of the superior wavelet is at least 10 percent higher than the inferior one

15 A first hint for the optimality of the four-staged WT has been given in the analysis about feature-extractors in section. Inan et al. [36] demonstrates the superiority of the fourth stage of a SWT approach
The first criterion is assessed using the signed-rank test according to Wilcoxon for paired samples (significance level \( p < 0.001 \)). The second one is evaluated by calculating the median of \( CR(\alpha, \theta_1) - CR(\alpha, \theta_2) \).

**Four stages are optimal** One major constraint is the variable \( \alpha \). Ten percent of the variance (\( \alpha = 0.1 \)) is easily covered by a small number of wavelet coefficients. Therefore, small values of \( \alpha \) yield high compression ratios. The figures below depict the compression ratio for \( \alpha = 0.5 \) and \( \alpha = 0.75 \) for 22 different wavelets. The striking element of the plots is the three ring structure corresponding to two-staged (red), three-staged (blue) and four-staged (black) wavelet decomposition. This is not surprising. Consider how wavelet transformation splits the signal into frequency bands. The two-staged setup consists of three bands (0-90Hz, 90-180, 180-360Hz). Its first sub-band roughly covers the range of the QRS-complex (4 22.5Hz). Hence, low compression is achieved. The three-staged setup again splits this first band into two sub-bands (0-45, 45-90Hz), which still covers too much non-QRS-information. The four-staged decomposition is quite optimal as it generates a new sub-band split (0-22.5, 22.5-45Hz), which is just the desired range of interest. The decimation process, enforced by the down-samplers of the WT, reduces the coefficients in the first sub-band, which contains the most variance. That is why the four-staged WT is optimal.

**Asymmetries** One verifies a strong asymmetry in the compression ratio of the different arrhythmias. The RBBB and SVES beats can only be compressed to 40 wavelet coefficients (median over all beats of that type). The normal beats on the other hand can be represented by 20 coefficients only. This asymmetry gets less pronounced in the three- (20 to 11 coeffs.) and four-(11 to 7 coeffs.) staged decomposition. The NPREM heartbeat shows another disparity, which is present at all stages. As it is highly concentrated in time, only a small amount of wavelet coefficients is needed to represent it.

These asymmetries fade as \( \alpha \) rises. If the wavelet coefficients have to cover increasing variance (higher \( \alpha \)), the idiosyncrasies of the beat types are less dominating. Hence, the amount of noise becomes more and more important. Given \( \alpha = 0.99 \) (see picture below). The shape of the rings become more circle-like, which shows the diminishing effect of the increasing noise contributions.

**Best wavelet can not be determined** Let us take a closer look at the inner circle: the wavelets used for four-staged WT decomposition. Maybe there is a substructure, which allows one to determine an optimal wavelet. The figures below depict the coif2, db2, db6 and sym6 wavelet. The coif2 was chosen from the other 16 four-staged wavelets, as it gives the best compression in most of the heartbeat types. The other three are the prevailing wavelets used for ECG-analysis. Both plots do not show the median number of wavelet coefficients only. They also give a glimpse into the width of the distribution of the wavelet coefficients for the instances of the corresponding beat types. Where \( \alpha = 0.9 \), the colored areas represent the interquartile width, i.e. the range between the lowest and highest quartile of the distribution of the wavelet coefficients. For \( \alpha = 0.99 \), the width of the lines is adjusted to cover the interquartile-width. In both scenarios the overlap between the wavelets is huge. Even though the overlap is a little reduced for \( \alpha = 0.99 \), this hardly has any consequences. The performance of wavelets for higher \( \alpha \) is mainly determined by the capability to represent the noise in the signal, rather than the information contained within it.

**Quantification** The results above need some quantification, even though the plots speak for themselves. I analyzed the compression capabilities of the sixteen wavelets using four-staged decomposition...
Figure 7.18: The median number of wavelet coefficients accounting for 50\% (top) and 75\% (bottom) of the variance for all thirteen arrhythmias and for 2 (red), 3 (blue) and 4 (black) wavelet decomposition stages.
Figure 7.19: The median number of wavelet coefficients accounting for 90\% (top left) and 99\% (top right) of the variance for all thirteen arrhythmias for 2 (red), 3 (blue) and 4 (black) wavelet decomposition stages. The plots below depict a detailed view of the four-staged wavelet decomposition showing the interquartile range of the number of wavelet coefficients accounting for 90\% (bottom left) 99\% (bottom right)
Chapter 7 Results

for $\alpha \in \{0.1, 0.25, 0.5, 0.75, 0.9, 0.99\}$. A wavelet was considered superior over another in case the following requisites held:

- the compression ratio of wavelet "A" is higher than the one in wavelet "B" in 75% of the instances of a certain heartbeat type
- equivalently, the upper quartile of the compression ratio of "A" should be less than the lower quartile of "B"
- superiority should exist in seven of thirteen heartbeat types
- the distance between the medians should be greater one.

The result of our analysis is that superior wavelets exist for $\alpha = 0.99$ only. The plot below shows two groups of wavelets that are distinct from each other by means of the above requisites. In fact, $\alpha = 0.99$ corresponds to the situation that the wavelets are "fitting" the noise only, rather than the idiosyncrasies of the signal.

![Figure 7.20: The median number of wavelet coefficients accounting for 99% variance](image-url)

Figure 7.20: The median number of wavelet coefficients accounting for 99% of the variance for all thirteen arrhythmias of a four-staged wavelet decomposition grouped into best (red), intermediate (black) and worst (blue) wavelets.
Chapter 8

Conclusion

The main objective of this thesis is to improve current ECG-classification methods. Thus, one major concern was to first determine the capabilities of current ECG-classifiers. This thesis provides the most extensive comparative study on ECG-classifiers in currently accessible English literature. Seventy classifiers have been not only investigated by means of performance, but additionally by assessing the whole setup of the classifier using the novel $\rho$-Score. It was developed during this thesis and rates the database usage, database coverage, amount of features, data and training data used. It reveals the current frontier of research in the field which each new classifier has to transcend.

The final classifier developed in this thesis accomplishes just that. In order to do so, new ways of feature extraction, feature selection and multi-class classification had to be found. The preprocessing stage of this thesis’ classifier follows a standard state-of-the-art approach in order to increase comparability of feature extraction and feature selection techniques on typical preprocessing grounds.

The feature extraction section provides the most extensive and comprehensive summary of feature extraction techniques towards the ECG in the last 20 years. 1.4 million features have been calculated, which cover not only the current area of research, but extend it with static delineation methods, more advanced segmentation methods with regard to the bi-spectrum and a wide range of relational features never employed so far. The powerful feature selection techniques developed in this thesis allow the analysis of the performance of the different basic feature extraction techniques. It supersedes a recent comparative study carried out by deLannoy et al. [42] by means of the number of features and feature-extractors under investigation.

The feature selection is the heart of this thesis. The new methods developed within this thesis allow the computation of small subsets (20-60 features) in reasonable time and with high performance. Six new measures of effect-size, the dichotomized effect-size indices (DESI), have been developed. Clearly, effect-size is only measured between two classes (group A and group B). Four methods have been engineered to combine these effect-size measures in a multi-class scenario with 13 classes. Those methods in combination with the DESI are all superseding the common FisherLDA feature selection technique. The best method, TopPCCorr(RiskEffect), is used for the final classifier.

The core classifier of this thesis is an ensemble of support vector machines (SVM). The number of SVMs that had to be trained for all experiments during the course of the thesis easily exceeds 200,000. As for each SVM an optimal configuration has to be determined, the number of configurations should be limited. This thesis provides a new method for doing so. It is based on the proposition that the optimal Gauss-kernel parameter $\gamma$ is equivalent to the optimal kernel-band-width of the histogram of Euclidean inter-sample distances. This proposition has been theoretically motivated and practically verified in binary and multi-class classification problems. The proposed approach uses less configuration than other schemes of configuration reduction while retaining high performance.
8.1 The current frontier of research

The biggest problem for defining a frontier in current research is that most researchers do not share the same setup or even present the same error measures for their classifiers. This thesis provides a way to rate each setup by means of quantitative and qualitative criteria: the $\rho$-Score. The higher the $\rho$-Score, the higher the standards of the framework in which the classifier performance was calculated. This thesis reveals that the $\rho$-Score and the performance measured via the $\kappa$-index are anti-correlated, i.e. higher performances are most likely to be gained in the case that the setup is quite sloppy (low $\rho$-Score): supervised record selection, too many features used, too much training data, too little database coverage. For the first time, it is now possible to rate a new classifier in an objective and dispassionate manner taking into account performance and classifier setup. My analysis, moreover, reveals a frontier of research each new classifier has to transcend. This frontier is invisible in studies mainly focusing on the accuracies given in the associated publications, e.g. Salem et al. [288].

The main limitation of this work is that the $\rho$-Score is quite subjective with regard to both scoring criteria and actual scores. The best way to create such a score would be to reach out for a wide consensus by many leading researchers in the field. The score provided here can only be a starting point for a discussion about rating the framework in which classification is carried out. Moreover, this thesis has used information in the associated publications only. Maybe some approaches would gain higher ratings providing that more information had been documented, e.g. the full confusion matrix. The author hopes that this review might lead to the usage of the proposed rating scheme in future ECG research and the development of a consensus-based rating system.

Even though the final classifier proposed in section 6.4 does improve the current frontier of research, it is not the final solution. It is still not good enough and still too close to other approaches. The main reason is supposedly the following: The database consists of annotations which are a consensus of cardiologists. This can cause a severe problem when facing noisy beats. The majority of mis-classifications of the proposed classifier are caused by non-normal beats that are detected as normal ones and normal beat that are assumed to be non-normal. One might speculate that noisy normal beats in the training set extend the “normal-space”, which leads to false normal classifications of non-normal beats. Noisy normal beats in the test set on the other hand will be accounted for non-normal ones leading to false non-normal classifications.

Future work has, therefore, to account for the different views cardiologists and computer algorithms have on the data. It might be true that a certain normal beat is indeed a normal beat. This information is useless if it is not a typical normal beat. An atypical normal beat should be marked as such or considered as an “alarm”-beat. For the next centuries, computers will only serve humans in a semi-automated scheme. Cardiologists have to decide which heart beat should be marked for human-revision.

8.2 Modern feature extraction

In the last 20 years, thousands of features describing the ECG have been devised. It would be great to bring all those features together, to test each of them and combine them to the most powerful feature set possible. Unfortunately, programming each single feature is nearly impossible. During the process of reviewing ECG classifiers it became obvious that the majority of features follows a simple scheme: segmentation-transformation-partitioning-assessment.

This thesis is the first to use seven different ECG-segmentation (most are centered around the R-peak)
strategies in order to focus on different aspects of the heartbeat. The subsequent transformation can be as simple as an identity-map (which means no transform at all) or as complex as wavelet transformations or the calculation of the bi-spectrum. The partitioning step again splits the transformed data in several parts of interest. The typical partitioning approach throughout the literature is a physiologic one. This means that the partition is chosen to depend on physiologic points of interest, like the QT or ST-segment of the QRS-complex. Even though this approach is sound and intuitive, one might wonder how those points should be determined. In fact, the estimation of the PQRST-points is quite cumbersome and results are quite noisy. Therefore, this thesis investigated another strategy: the symmetric or static partitioning approach. Suppose one wants to analyze the P-wave the QRS-complex and the T-wave.

The roughest estimate would be to split the ECG-segment into 3 partitions of equal size. The left part contains the P-wave and the center the QRS-complex. One can go further by splitting it into even-numbered partitions in order to analyze the slopes of the QRS-complex. In the perspective of partitioning, the wavelet decomposition performs transformation and partitioning at the same time. It splits the data into chunks of different time-resolution and frequency content. Finally, each partition can be assessed using 22 partition-functions like median, variance or mean value. This scheme allows the calculation of the great majority of features used in the literature. Together with RR-interval parameters, new feature extraction methods towards the bi-spectrum, the novel static delineation approach and an array of relational features, a database of 1.4 million features that have never been tested before, has been composed. In combination with new tools of feature selection, it is now possible to group those features according to the basic feature extraction techniques they are derived from. Thus, this thesis presents the most extensive comparative study on feature extraction techniques for ECG classification, superseding previous attempts by [42].

One result of this investigation is that measures of static wavelet transform (SWT) yield a higher performance than those derived from the discrete wavelet transform (DWT). It shows that the third-order Cumulants contain viable information and that auto-regressive models can not compete with other feature-extractors. The power of RR-Interval features is also limited. They contribute two to four percent points to the \( jk \)-Index only in cases where features are used that are based solely on QRS-morphology information or on physiologic delineation. In all other scenarios they contribute less than one percent point.

The main advantage of this thesis is that the feature selection technology can take care of a vast amount of features. Previous studies by [38, 42] could only group several features by some feature-extractors summing up to a 50 features under investigation. This thesis analyzes over 1.4 million. As this thesis tries to classify thirteen different ECG-morphologies, a feature-extractor has to account for many morphologies and so forth. If only four or five classes are used, (see [38, 42]) feature-extractors that are capable of selecting only those morphologies are favored, not those which allow a flexible characterization of the ECG. Moreover, using more classes reduces the impact of the classification performance towards normal beats.

The main limitation of this thesis is that the analysis is based on a single ECG-database and a single lead. As discussed in section 2.2, the current databases do not allow a direct cross-validation of the exact same lead and with a comparable distribution of arrhythmias. Moreover, one should be aware that the size of the feature groups, which account for certain feature-extractors and are compared in section 7.3, span over several magnitudes (1000 to 100000). Even though the correlation between feature group size and \( jk \)-Index is limited, it is not negligible. One might argue that a very flexible feature-extractor naturally yields more possibilities for features and therefore leads to larger feature groups.

Future work must validate my findings on further leads and databases and even more data. Furthermore, it should include Hermite polynomials and Fourier spectrum estimators as feature-extractors to show their inferiority in a comparative study.
Chapter 8 Conclusion

Use of wavelets This thesis demonstrates that optimal wavelet decomposition can be obtained in case the wavelet is furnished to analyze the frequency-band between 2 and 20Hz, which covers the QRS-frequency range. The majority of researchers claim a 4-staged wavelet decomposition to be optimal. This is quite true. The sampling frequency of most databases like MIT-BIH Arrhythmia or Longterm-ST database resides between 150 to 500Hz. In this situation, one sub-band of the 4-staged decomposition overlaps with the QRS-frequency range. This thesis shows that a four staged decomposition of orthogonal wavelets has the highest compression ratio, i.e. a small number of wavelet coefficients represents most of the variance in the ECG. This is also backed by the fact that the final feature set is comprised predominantly by features relying on the fourth level of a static wavelet transform. Again, the fourth level has the biggest overlap with the QRS-frequency range. With regard to the compression ratio, it is of no importance which orthogonal wavelet family is chosen. All 22 wavelets under investigation showed no significant difference in compression ratio. Only if wavelets should represent more than 99% of the signals variance, differences arise. Those differences are not that important because the remaining 1% is mainly noise and it is surely not the goal of optimize for the noise in the signal.

If the only importance of wavelet decomposition is the overlap with the QRS-spectrum, I would recommend to use the static wavelet transform, as the time-resolution is quite higher than in the orthogonal case.

8.3 Fast and high performing multi-class feature selection

The major task in this thesis was to select 20-60 features from the vast amount of 1.4 million. This amount renders typical wrapper and embedded methods of feature selection infeasible. Thus, filters or feature ranking methods had to be employed. Unfortunately, the majority of feature ranking techniques is based on parametric or correlation-based measures of effect-size. As assumptions about the underlying distributions of each of the 1.4 million features should be avoided, this road seems like a dead-end. More powerful feature filters have been developed in the text-categorization domain. Suppose one has to find all important websites concerning the category "Sports in Germany". Now, one searches for certain words in those websites and measures the likelihood of a website containing this word to be in that category. This seems quite the same situation as in ECG-classification, yet only quite. A certain word either being in or not in a document is a binomial problem. The features for ECG-classification on the other hand are of cardinal scale.

Thus, this thesis transferred measures established in the text-categorization domain and extended them to rate cardinal instead of binary or dichotomized data. These dichotomized effect-size indices (DESI) showed superior power, both in Toy-Monte-Carlo and ECG feature selection experiments. The final challenge was not only to rate a feature towards a binary classification task, but towards a multi-class scenario with 13 different classes. Four methods have been proposed and compared to the standard FisherLDA procedure rendering the latter inferior. It could also be demonstrated, that the Fisher-Score is inferior no matter which multi-class combination scheme is applied. This reveals that parametric effect-size indices should not be considered for this amount of data when no assumptions can be made about the underlying distributions.

The applicability of the final TopPCCorr(RiskEffect) feature selection has yet to be proven in other domains. Even though Toy-Monte-Carlo experiments show the superiority of DESI over standard effect-size indices, other high-dimensional multi-class databases have to be tested to generalize my findings. Another strong hint for DESI being generally superior is that the decision in favour of the DESI has been made with only a tenth of the feature database at hand. Moreover, the DESI approach has been applied to smaller portions of the database for the purpose of feature-extractor comparisons. In all those
8.4 New insights in the application of support vector machines

The NIPS challenge 2003 "revealed that hyperparameter selection may have played an important role on winning the challenge. Indeed, several groups were using the same classifier (e.g. a SVM) and reported significantly different results. We have started laying the basis of a new benchmark on the theme of model selection and hyperparameter selection."[178] This quote from 2003 shows the importance of configuring the SVM right. The choice of hyperparameters, the amount of data used, the normalization employed, the SVM ensemble method and the kernel applied are crucial for the final result.

A large study by [15] extensively studied SVM ensemble methods showing that no one ensemble method is superior to one another. Their major limitation was that they employed a polynomial kernel with fixed hyperparameters. My analysis showed that in the case that an elaborate hyperparameter optimization is carried out, ensemble methods do matter. Dense error correcting output codes (ECOC) and Sparse ECOC are on average significantly better than One vs. One (OVO) or One vs. All (OVA) schemes. Even though the average performance of ECOC is higher than for OVO and OVA, this is not true for all datasets. The major problem is that in multi-class scenarios above 7 classes suboptimal ECOCs have to be employed to ensure computational feasibility. Those ECOCs are asymmetric, i.e. certain classes are more often on the "left" side of the binary SVM problem than on the "right" side. This can be either beneficial or detrimental depending on the data at hand. That is why future work will concentrate on creating data-dependent ECOC schemes. Promising work in this direction has been carried out by [281–284]. Another advantage of ECOCs not yet discussed is their robustness towards smaller data-amounts. This thesis could show that ECOCs retain a higher performance when facing lower amounts of data than the OVO and OVA scheme. In this respect, ECOCs do a better job in generalizing the structure of the data.

Throughout the literature little attention has been paid to the normalization of the data. This thesis tested eleven different normalizations and revealed that six normalizations yield high performance and do not significantly differ from each other. One of those six is the standard \[0, \ldots, 1\] normalization. So, the little attention was partially justified. However, for some datasets performance can increase up to 3-5 %depending on the normalization. In the future, further research should be carried out to develop a system which automatically assigns the best normalization method for the data at hand. This would supposedly mean normalizing each feature with another normalization approach. Still, it is unknown what the optimal normalization would look like given many features are normally distributed and others which are exponentially distributed.

The data sets and data samples from [15] also allowed one to test if the new hyperparameter search developed in this thesis really does show good results. The results indicate that the OVO and OVA scheme is indeed faster and retains the same performance. The performance of ECOCs drops by one or two percent only. In conclusion, this faster search scheme allows the usage of SVMs not only as a final classifier, but as a handy tool which can report results in minutes rather than hours. Moreover, it demonstrates that the hyperparameters are not abstract. Instead, one can now split what was previously a two-dimensional grid search into two one-dimensional optimization problems. In situations where parallel-computing is not available, one can even use a decent hill-climbing algorithm to reduce configuration testing further.

More generally spoken, the success of this approach towards Gaussian kernels raises the question if the optimal choice of a kernel for non-linear SVMs is in fact the kernel which optimizes the histogram of
inter-sample distances best. This is, of course, only possible for kernels which make use of the inter-sample distance. Moreover, it will be a challenge to calculate the optimal bandwidth of other kernels as fast as the one of the Gaussian. It is only one percent of the calculation time of the SVM. This is quite doubtable because the high-dimensional Gaussian can be factorized into multiple one-dimensional ones, which eases calculation.
Appendices
Appendix A

The final feature sets

This appendix will show the medium-sized (25) and large (49) feature set with all its features. See 7.4 for information on the syntax of the features shown below.

A.1 The TopPCCorr(2, 50) dataset (only SWT features)

The feature set consists of the following features:

\[
\begin{align*}
RRRel5 & \quad (A.1) \\
RRInst & \quad (A.2) \\
Y(\Delta_{seg} = 300, MMNorm) \rightarrow SWT(1) & \rightarrow AvgPower_{16}^9 \quad (A.3) \\
& \rightarrow Min_{5}^3 \quad (A.4) \\
& \rightarrow TrendCut_{3}^2 \quad (A.5) \\
& \rightarrow AbsMean_{8}^5 \quad (A.6) \\
& \rightarrow Min_{4}^4 \quad (A.7) \\
SWT(3) & \rightarrow TrendSlope_{1}^1 \quad (A.8) \\
& \rightarrow TrendSlope_{8}^1 \quad (A.9) \\
Y(\Delta_{seg} = 150, MMNorm) \rightarrow SWT(1) & \rightarrow Max_{16}^9 \quad (A.10) \\
& \rightarrow Min_{3}^3 \quad (A.11) \\
& \rightarrow TrendSlope_{3}^3 \quad (A.12) \\
& \rightarrow Min_{8}^4 \quad (A.13) \\
SWT(2) & \rightarrow SWT(3) \rightarrow MinMaxAbs_{3}^1 \quad (A.14) \\
& \rightarrow Mean_{5}^5 \quad (A.15) \\
SWT(4) & \rightarrow SWT(2) \rightarrow TrendSlope_{3}^3 \quad (A.16) \\
& \rightarrow TrendSlope_{3}^3 \quad (A.17) \\
Y(\Delta_{seg} = 300, NONorm) \rightarrow SWT(2) & \rightarrow MiMaxAbs_{8}^4 \quad (A.18) \\
& \rightarrow SWT(3) \rightarrow TrendCut_{5}^2 \quad (A.19) \\
& \rightarrow SWT(4) \rightarrow MaxAbs_{3}^3 \quad (A.20)
\end{align*}
\]
Appendix A The final feature sets

\[
\begin{align*}
Y(\Delta_{seg} = 300, MedNorm) \rightarrow SWT(4) & \rightarrow XX_5^4 \quad (A.21) \\
Y(\Delta_{seg} = 150, MedNorm) \rightarrow SWT(3) & \rightarrow \text{AbsMean}_{3}^3 \quad (A.22) \\
\text{SWT}(4) & \rightarrow \text{MiMax}_{8}^4 \quad (A.23) \\
Y(\Delta_{seg} = 250') \rightarrow SWT(4) & \rightarrow \text{AbsMean}_{16}^4 \quad (A.24) \\
\end{align*}
\]

A.2 The TopPCCorr(4, 0.5) feature-set (only SWT features)

The feature set consists of features in A.1, A.2, A.8, A.10, A.12, A.13, A.15, A.17, A.20 and those defined below:

\[
\begin{align*}
\text{Prem} \\
Y(\Delta_{seg} = 300, MMNorm) \rightarrow SWT(2) & \rightarrow \text{MiMax}_{16}^6 \quad (A.26) \\
& \rightarrow \text{Var}_{16}^6 \quad (A.27) \\
& \rightarrow \text{Min}_{8}^4 \quad (A.28) \\
\text{SWT}(3) & \rightarrow \text{StdDev}_{16}^6 \quad (A.29) \\
& \rightarrow \text{Var}_{16}^6 \quad (A.30) \\
& \rightarrow \text{TrendCut}_{3}^2 \quad (A.31) \\
& \rightarrow \text{TrendSlope}_{3}^3 \quad (A.32) \\
\text{SWT}(4) & \rightarrow \text{MaxAbs}_{3}^2 \quad (A.33) \\
& \rightarrow \text{MiMax}_{3}^2 \quad (A.34) \\
& \rightarrow \text{StdDev}_{8}^5 \quad (A.35) \\
& \rightarrow \text{Var}_{8}^5 \quad (A.36) \\
Y(\Delta_{seg} = 150, MMNorm) \rightarrow SWT(1) & \rightarrow \text{Min}_{3}^2 \quad (A.37) \\
& \rightarrow \text{Min}_{5}^3 \quad (A.38) \\
\text{SWT}(2) & \rightarrow \text{TrendCut}_{16}^6 \quad (A.39) \\
\text{SWT}(3) & \rightarrow \text{Min}_{16}^5 \quad (A.40) \\
\text{SWT}(4) & \rightarrow \text{TrendSlope}_{8}^4 \quad (A.41) \\
Y(\Delta_{seg} = 300, NONorm) \rightarrow SWT(3) & \rightarrow \text{MiMax}_{16}^4 \quad (A.42) \\
\text{SWT}(4) & \rightarrow \text{MiMaxAbs}_{16}^6 \quad (A.43) \\
& \rightarrow \text{MiMax}_{3}^1 \quad (A.44) \\
& \rightarrow \text{StdDev}_{3}^1 \quad (A.45) \\
& \rightarrow \text{TrendSlope}_{3}^3 \quad (A.46) \\
\end{align*}
\]
A.2 The TopPCCorr(4, 0.5) feature-set (only SWT features)

\[ Y(\Delta_{seg} = 150, NONorm) \rightarrow SWT(2) \rightarrow \]
\[ SWT(3) \rightarrow \]
\[ SWT(4) \rightarrow \]
\[ Y(\Delta_{seg} = 150, MedNorm) \rightarrow SWT(2) \rightarrow \]
\[ SWT(3) \rightarrow \]
\[ SWT(4) \rightarrow \]
\[ Y(\Delta_{seg} = 250') \rightarrow SWT(3) \rightarrow \]
\[ SWT(4) \rightarrow \]

\[ AbsMean^2 \] (A.50)
\[ Max^2 \] (A.51)
\[ Mean^3 \] (A.52)
\[ Max^4 \] (A.53)
\[ Max^5 \] (A.54)
\[ MiMax^4 \] (A.55)
\[ Entropy^3 \] (A.56)
\[ AvgPower^4 \] (A.57)
\[ MiMax^4 \] (A.58)
\[ MiMaxAbs^4 \] (A.59)
\[ MiMaxAbs^4 \] (A.60)
\[ MaxAbs^{5,6} \] (A.61)
\[ TrendSlope^5 \] (A.62)
\[ Mean^5 \] (A.63)
\[ MaxAbs^{1,6} \] (A.64)
\[ Min^2 \] (A.65)
Appendix B

The confusion matrices

This appendix shows the mean confusion matrices of the three final feature sets for 10 training-test splits. The data was split according to section 2.2.1. Each confusion matrix is accompanied with the sample error estimate of each entry in the confusion matrix presented in a second “error”-confusion-matrix.
### Appendix B The confusion matrices

<table>
<thead>
<tr>
<th>N</th>
<th>VP</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>55397.8</td>
<td>1.6</td>
<td>171.5</td>
<td>260.3</td>
<td>1.0</td>
<td>49.3</td>
<td>44.4</td>
<td>21.8</td>
<td>54.2</td>
<td>8.0</td>
<td>0.7</td>
</tr>
<tr>
<td>vP</td>
<td>6.4</td>
<td>1802.7</td>
<td>2.9</td>
<td>0.0</td>
<td>3.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VPC</td>
<td>164.4</td>
<td>0.0</td>
<td>4.8</td>
<td>942.1</td>
<td>0.0</td>
<td>130.9</td>
<td>5.4</td>
<td>1.1</td>
<td>4.8</td>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>APC</td>
<td>164.4</td>
<td>0.0</td>
<td>4.8</td>
<td>942.1</td>
<td>0.0</td>
<td>130.9</td>
<td>5.4</td>
<td>1.1</td>
<td>4.8</td>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>FPN</td>
<td>2.8</td>
<td>1.5</td>
<td>1.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>RBBB</td>
<td>39.4</td>
<td>0.0</td>
<td>3.2</td>
<td>50.2</td>
<td>0.0</td>
<td>4638.3</td>
<td>0.2</td>
<td>0.3</td>
<td>1.6</td>
<td>5.6</td>
<td>0.4</td>
</tr>
<tr>
<td>LBBB</td>
<td>54.4</td>
<td>0.0</td>
<td>33.5</td>
<td>17.7</td>
<td>0.0</td>
<td>5405.1</td>
<td>0.9</td>
<td>1.8</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>FVN</td>
<td>59.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>AAP</td>
<td>4.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NESC</td>
<td>12.7</td>
<td>0.0</td>
<td>3.1</td>
<td>10.0</td>
<td>0.0</td>
<td>379</td>
<td>6.0</td>
<td>0.0</td>
<td>3.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NPREM</td>
<td>0.0</td>
<td>0.0</td>
<td>4.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>OTHER</td>
<td>0.0</td>
<td>0.0</td>
<td>4.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Total</td>
<td>55911.0</td>
<td>18050.0</td>
<td>4728.0</td>
<td>1266.0</td>
<td>130.0</td>
<td>4831.0</td>
<td>5372.0</td>
<td>401.0</td>
<td>75.0</td>
<td>115.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error of confusion matrix entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
</tr>
<tr>
<td>vP</td>
</tr>
<tr>
<td>VPC</td>
</tr>
<tr>
<td>APC</td>
</tr>
<tr>
<td>FPN</td>
</tr>
<tr>
<td>RBBB</td>
</tr>
<tr>
<td>LBBB</td>
</tr>
<tr>
<td>FVN</td>
</tr>
<tr>
<td>AAP</td>
</tr>
<tr>
<td>NESC</td>
</tr>
<tr>
<td>NPREM</td>
</tr>
<tr>
<td>OTHER</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Table B.1: Mean and error estimate of the confusion matrix of the TopPCCorr(1, 0.5) feature set (see section 7.4) averaged over 10 training-test splits with ratios according to subsection 2.2.1.
<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>vP</th>
<th>VPC</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>55582.2</td>
<td>0.7</td>
<td>78.8</td>
<td>106.1</td>
<td>0.7</td>
<td>14.7</td>
<td>28.8</td>
<td>39.2</td>
<td>11.3</td>
<td>22.6</td>
<td>3.7</td>
<td>2.4</td>
<td>18.0</td>
</tr>
<tr>
<td>vP</td>
<td>4.6</td>
<td>1801.8</td>
<td>0.6</td>
<td>0.0</td>
<td>3.4</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>VPC</td>
<td>72.1</td>
<td>0.0</td>
<td>4577.0</td>
<td>4.6</td>
<td>0.0</td>
<td>5.1</td>
<td>13.0</td>
<td>27.1</td>
<td>5.6</td>
<td>0.0</td>
<td>1.7</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>APC</td>
<td>133.0</td>
<td>0.0</td>
<td>11.4</td>
<td>1149.0</td>
<td>0.0</td>
<td>6.0</td>
<td>4.2</td>
<td>0.8</td>
<td>3.7</td>
<td>11.1</td>
<td>3.0</td>
<td>0.0</td>
<td>0.6</td>
</tr>
<tr>
<td>FPN</td>
<td>2.5</td>
<td>2.3</td>
<td>0.4</td>
<td>0.0</td>
<td>125.8</td>
<td>0.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>RBBB</td>
<td>8.8</td>
<td>0.0</td>
<td>1.5</td>
<td>43.0</td>
<td>0.0</td>
<td>4803.3</td>
<td>0.4</td>
<td>0.9</td>
<td>0.1</td>
<td>1.5</td>
<td>1.2</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>LBBB</td>
<td>25.3</td>
<td>0.0</td>
<td>11.4</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>5322.5</td>
<td>1.7</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>FVN</td>
<td>34.1</td>
<td>0.0</td>
<td>36.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.5</td>
<td>1.6</td>
<td>330.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
</tr>
<tr>
<td>AAP</td>
<td>2.7</td>
<td>0.0</td>
<td>5.7</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>53.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NESC</td>
<td>29.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>88.2</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NPREM</td>
<td>7.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.6</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.7</td>
<td>0.0</td>
<td>1.5</td>
<td>36.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VESC</td>
<td>0.0</td>
<td>0.0</td>
<td>1.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>48.4</td>
<td>0.6</td>
</tr>
<tr>
<td>OTHER</td>
<td>8.9</td>
<td>0.2</td>
<td>3.3</td>
<td>0.0</td>
<td>0.1</td>
<td>0.8</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>15.7</td>
</tr>
<tr>
<td>Total</td>
<td>55911.0</td>
<td>1805.0</td>
<td>4728.0</td>
<td>1266.0</td>
<td>130.0</td>
<td>4831.0</td>
<td>5372.0</td>
<td>401.0</td>
<td>75.0</td>
<td>115.0</td>
<td>42.0</td>
<td>53.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

Error of confusion matrix entries

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>vP</th>
<th>VPC</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>11.2</td>
<td>0.4</td>
<td>6.1</td>
<td>5.0</td>
<td>0.2</td>
<td>1.6</td>
<td>2.1</td>
<td>3.1</td>
<td>1.7</td>
<td>1.9</td>
<td>0.4</td>
<td>0.4</td>
<td>0.7</td>
</tr>
<tr>
<td>vP</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.0</td>
<td>0.9</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>VPC</td>
<td>5.0</td>
<td>0.0</td>
<td>8.4</td>
<td>0.4</td>
<td>0.0</td>
<td>1.1</td>
<td>1.7</td>
<td>1.2</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>APC</td>
<td>10.5</td>
<td>0.0</td>
<td>1.8</td>
<td>5.4</td>
<td>0.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.2</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>FPN</td>
<td>0.4</td>
<td>0.5</td>
<td>0.2</td>
<td>0.0</td>
<td>0.9</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>RBBB</td>
<td>0.9</td>
<td>0.0</td>
<td>0.3</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.1</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>LBBB</td>
<td>1.6</td>
<td>0.0</td>
<td>1.7</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>FVN</td>
<td>4.2</td>
<td>0.0</td>
<td>2.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>AAP</td>
<td>0.4</td>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NESC</td>
<td>2.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>1.6</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NPREM</td>
<td>1.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.3</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VESC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>OTHER</td>
<td>0.7</td>
<td>0.1</td>
<td>0.3</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>Total</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table B.2: Mean and error estimate of the confusion matrix of the TopPCCorr(2, 0.5) feature set (see section A.1) averaged over 10 training-test splits with ratios according to subsection 2.2.1.
## Appendix B: The confusion matrices

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>vP</th>
<th>VPC</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>55649.5</td>
<td>0.4</td>
<td>67.4</td>
<td>101.3</td>
<td>0.4</td>
<td>13.3</td>
<td>17.6</td>
<td>42.4</td>
<td>7.7</td>
<td>22.8</td>
<td>2.7</td>
<td>1.0</td>
<td>16.5</td>
</tr>
<tr>
<td>vP</td>
<td>1.2</td>
<td>1801.9</td>
<td>0.2</td>
<td>0.0</td>
<td>3.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VPC</td>
<td>52.6</td>
<td>0.0</td>
<td>4587.4</td>
<td>3.6</td>
<td>0.0</td>
<td>4.9</td>
<td>13.6</td>
<td>24.3</td>
<td>6.1</td>
<td>0.0</td>
<td>1.6</td>
<td>9.9</td>
<td>9.7</td>
</tr>
<tr>
<td>APC</td>
<td>113.2</td>
<td>0.0</td>
<td>8.7</td>
<td>1150.0</td>
<td>6.5</td>
<td>1.8</td>
<td>1.2</td>
<td>2.9</td>
<td>0.9</td>
<td>0.4</td>
<td>0.2</td>
<td>0.9</td>
<td>9.7</td>
</tr>
<tr>
<td>FPN</td>
<td>1.9</td>
<td>2.7</td>
<td>0.3</td>
<td>0.0</td>
<td>126.3</td>
<td>0.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>9.7</td>
</tr>
<tr>
<td>RBBB</td>
<td>5.0</td>
<td>0.0</td>
<td>1.1</td>
<td>9.2</td>
<td>0.0</td>
<td>4804.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.0</td>
<td>1.4</td>
<td>1.3</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>LBBB</td>
<td>11.5</td>
<td>0.0</td>
<td>10.2</td>
<td>0.0</td>
<td>0.1</td>
<td>0.5</td>
<td>5335.9</td>
<td>0.6</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>9.7</td>
</tr>
<tr>
<td>FVN</td>
<td>35.6</td>
<td>0.0</td>
<td>41.2</td>
<td>0.2</td>
<td>0.0</td>
<td>0.3</td>
<td>1.1</td>
<td>331.7</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>9.7</td>
</tr>
<tr>
<td>AAP</td>
<td>2.7</td>
<td>0.0</td>
<td>6.7</td>
<td>0.2</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>57.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>NESC</td>
<td>27.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.3</td>
<td>0.0</td>
<td>0.9</td>
<td>0.3</td>
<td>0.2</td>
<td>0.0</td>
<td>88.6</td>
<td>0.2</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>NPREM</td>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>0.0</td>
<td>1.3</td>
<td>37.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VESC</td>
<td>0.0</td>
<td>0.0</td>
<td>2.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.3</td>
</tr>
<tr>
<td>OTHER</td>
<td>9.7</td>
<td>0.0</td>
<td>2.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>13.9</td>
</tr>
<tr>
<td>Total</td>
<td>55911.0</td>
<td>1805.0</td>
<td>4728.0</td>
<td>1266.0</td>
<td>130.0</td>
<td>4831.0</td>
<td>5372.0</td>
<td>401.0</td>
<td>75.0</td>
<td>115.0</td>
<td>42.0</td>
<td>53.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>vP</th>
<th>VPC</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>4.9</td>
<td>0.2</td>
<td>3.0</td>
<td>4.8</td>
<td>0.2</td>
<td>1.2</td>
<td>0.9</td>
<td>2.1</td>
<td>0.9</td>
<td>1.4</td>
<td>0.3</td>
<td>0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>vP</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VPC</td>
<td>2.7</td>
<td>0.0</td>
<td>3.7</td>
<td>0.5</td>
<td>0.0</td>
<td>0.7</td>
<td>2.0</td>
<td>1.3</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>1.5</td>
</tr>
<tr>
<td>APC</td>
<td>4.3</td>
<td>0.0</td>
<td>0.9</td>
<td>5.7</td>
<td>0.0</td>
<td>1.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>FPN</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>0.0</td>
<td>0.6</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>RBBB</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>1.4</td>
<td>0.0</td>
<td>2.6</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>LBBB</td>
<td>1.2</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>2.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>FVN</td>
<td>2.6</td>
<td>0.0</td>
<td>2.9</td>
<td>0.1</td>
<td>0.0</td>
<td>0.2</td>
<td>0.4</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>AAP</td>
<td>0.6</td>
<td>0.0</td>
<td>0.7</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>NESC</td>
<td>2.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>1.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>NPREM</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>VESC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>OTHER</td>
<td>1.1</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Total</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>vP</th>
<th>VPC</th>
<th>APC</th>
<th>FPN</th>
<th>RBBB</th>
<th>LBBB</th>
<th>FVN</th>
<th>AAP</th>
<th>NESC</th>
<th>NPREM</th>
<th>VESC</th>
<th>OTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>55911.0</td>
<td>1805.0</td>
<td>4728.0</td>
<td>1266.0</td>
<td>130.0</td>
<td>4831.0</td>
<td>5372.0</td>
<td>401.0</td>
<td>75.0</td>
<td>115.0</td>
<td>42.0</td>
<td>53.0</td>
<td>42.0</td>
</tr>
</tbody>
</table>

Table B.3: Mean and error estimate of the confusion matrix of the TopPCCorr(4, 0.5) feature set (see section A.2) averaged over 10 training-test splits with ratios according to subsection 2.2.1.
## Appendix C

### The scoring of this thesis’ feature sets

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>author</strong></td>
<td>This Thesis</td>
<td>3/3</td>
</tr>
<tr>
<td><strong>ecg-db</strong></td>
<td>MIT</td>
<td>4/4</td>
</tr>
<tr>
<td><strong>ρd</strong></td>
<td>100%</td>
<td>3/8</td>
</tr>
<tr>
<td><strong>ρt</strong></td>
<td>28.8%(^a)</td>
<td>2/2</td>
</tr>
<tr>
<td><strong>selfunc</strong></td>
<td>unsupervised</td>
<td>5/5</td>
</tr>
<tr>
<td><strong>rf</strong></td>
<td>(\frac{13}{12} = 1.0; \ \left(\frac{25}{12} = 2.1\right))</td>
<td>4(4)[2][6]</td>
</tr>
<tr>
<td><strong>confmatrix</strong></td>
<td>Conf. with err.</td>
<td>0/2</td>
</tr>
<tr>
<td><strong>crossval-db</strong></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td><strong>unknown-beat</strong></td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td><strong>classes</strong></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td><strong>total points</strong></td>
<td>23(23)[21]/32</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) some beats with 25 %, some with 50 %, calculated total train-set/total test-set

Without using the unknown-beat one gains:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>author</strong></td>
<td>This Thesis</td>
<td>3/3</td>
</tr>
<tr>
<td><strong>ecg-db</strong></td>
<td>MIT</td>
<td>4/4</td>
</tr>
<tr>
<td><strong>ρd</strong></td>
<td>100%</td>
<td>3/8</td>
</tr>
<tr>
<td><strong>ρt</strong></td>
<td>28.8%(^a)</td>
<td>2/2</td>
</tr>
<tr>
<td><strong>selfunc</strong></td>
<td>unsupervised</td>
<td>5/5</td>
</tr>
<tr>
<td><strong>rf</strong></td>
<td>(\frac{13}{12} = 1.1; \ \left(\frac{25}{12} = 2.1\right))</td>
<td>4(2)[1][6]</td>
</tr>
<tr>
<td><strong>confmatrix</strong></td>
<td>Conf. with err.</td>
<td>0/2</td>
</tr>
<tr>
<td><strong>crossval-db</strong></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td><strong>unknown-beat</strong></td>
<td>Yes</td>
<td>0/2</td>
</tr>
<tr>
<td><strong>classes</strong></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td><strong>total points</strong></td>
<td>21(19)[17]/32</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) some beats with 25 %, some with 50 %, calculated total train-set/total test-set
## Appendix D

### Summary of reviewed publications

<table>
<thead>
<tr>
<th>Author</th>
<th>$\rho$</th>
<th>jx-Index</th>
<th>C</th>
<th>Author</th>
<th>$\rho$</th>
<th>jx-Index</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Song et al. [204]</td>
<td>28</td>
<td>69.40%</td>
<td>5</td>
<td>Park et al. [41]</td>
<td>9</td>
<td>90.00%</td>
<td>4</td>
</tr>
<tr>
<td>Tsipouras et al. [34]</td>
<td>26</td>
<td>66.90%</td>
<td>4</td>
<td>deLannoy et al. [42]</td>
<td>9</td>
<td>90.00%</td>
<td>4</td>
</tr>
<tr>
<td>Ince et al. [163]</td>
<td>24</td>
<td>69.40%</td>
<td>8</td>
<td>de Chazal and Reilly [83]</td>
<td>8</td>
<td>91.92%</td>
<td>3</td>
</tr>
<tr>
<td>Lagerholm et al. <a href="i">47</a></td>
<td>22</td>
<td>&lt;=9%</td>
<td>16</td>
<td>Chudácek et al. [24]</td>
<td>8</td>
<td>&lt;=9%</td>
<td>2</td>
</tr>
<tr>
<td>Shyu et al. [111]</td>
<td>19</td>
<td>85.49%</td>
<td>2</td>
<td>Pardey [139]</td>
<td>7</td>
<td>97.94%</td>
<td>2</td>
</tr>
<tr>
<td>Lai and Chan [85]</td>
<td>19</td>
<td>70.51%</td>
<td>2</td>
<td>Chi et al. [84]</td>
<td>7</td>
<td>94.65%</td>
<td>3</td>
</tr>
<tr>
<td>Rodriguez et al. [5]</td>
<td>18</td>
<td>88.82%</td>
<td>13</td>
<td>Coast et al. [222]</td>
<td>7</td>
<td>92.99%</td>
<td>2</td>
</tr>
<tr>
<td>Lagerholm et al. <a href="ii">47</a></td>
<td>18</td>
<td>92.41%</td>
<td>13</td>
<td>Abibullaev and Seo [105]</td>
<td>7</td>
<td>&lt;=9%</td>
<td>2</td>
</tr>
<tr>
<td>Jekova et al. [64]</td>
<td>18</td>
<td>&lt;=9%</td>
<td>5</td>
<td>Ge et al. [90]</td>
<td>7</td>
<td>&lt;=9%</td>
<td>6</td>
</tr>
<tr>
<td>Jiang and Kong <a href="i">60</a></td>
<td>18</td>
<td>69.84%</td>
<td>5</td>
<td>Acir [61]</td>
<td>7</td>
<td>&lt;=9%</td>
<td>6</td>
</tr>
<tr>
<td>Prasad and Sahambi [17]</td>
<td>17</td>
<td>&lt;=9%</td>
<td>13</td>
<td>Froese et al. [70]</td>
<td>6</td>
<td>&lt;=9%</td>
<td>2</td>
</tr>
<tr>
<td>Chang et al. [223]</td>
<td>16</td>
<td>&lt;=9%</td>
<td>5</td>
<td>Zhou [230]</td>
<td>6</td>
<td>&lt;=9%</td>
<td>2</td>
</tr>
<tr>
<td>Inan et al. [36]</td>
<td>16</td>
<td>96.90%</td>
<td>3</td>
<td>Özbay [231]</td>
<td>5</td>
<td>100.00%</td>
<td>10</td>
</tr>
<tr>
<td>Jekova et al. [62]</td>
<td>16</td>
<td>83.83%</td>
<td>2</td>
<td>Dokuz et al. [98]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>10</td>
</tr>
<tr>
<td>Zadeh et al. [68]</td>
<td>16</td>
<td>93.67%</td>
<td>3</td>
<td>Yu and Chen [118]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>2</td>
</tr>
<tr>
<td>de Chazal and Reilly <a href="i">292</a></td>
<td>16</td>
<td>68.46%</td>
<td>5</td>
<td>Osowskis and Linh [117]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>7</td>
</tr>
<tr>
<td>Bortolan et al. <a href="i">66</a></td>
<td>16</td>
<td>49.55%</td>
<td>2</td>
<td>Huang et al. [280]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>8</td>
</tr>
<tr>
<td>Jiang and Kong <a href="ii">60</a></td>
<td>16</td>
<td>77.50%</td>
<td>5</td>
<td>Barro et al. [22]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>15</td>
</tr>
<tr>
<td>Moraes et al. [138]</td>
<td>16</td>
<td>&lt;=9%</td>
<td>5</td>
<td>Hu et al. [30]</td>
<td>5</td>
<td>&lt;=9%</td>
<td>4</td>
</tr>
<tr>
<td>Moody and Mark [46]</td>
<td>16</td>
<td>&lt;=9%</td>
<td>2</td>
<td>Yu and Chou [110]</td>
<td>4</td>
<td>&lt;=9%</td>
<td>6</td>
</tr>
<tr>
<td>LLamendo and Martínez [21]</td>
<td>15</td>
<td>74.97%</td>
<td>3</td>
<td>Nazmy et al. [91]</td>
<td>4</td>
<td>97.31%</td>
<td>6</td>
</tr>
<tr>
<td>LLamendo and Martínez <a href="ii">40</a></td>
<td>15</td>
<td>74.97%</td>
<td>3</td>
<td>Guler and Übeyli [131]</td>
<td>3</td>
<td>&lt;=9%</td>
<td>4</td>
</tr>
<tr>
<td>Yeap [35]</td>
<td>14</td>
<td>77.31%</td>
<td>2</td>
<td>Acir [8]</td>
<td>3</td>
<td>96.88%</td>
<td>10</td>
</tr>
<tr>
<td>Mar et al. [7]</td>
<td>14</td>
<td>64.08%</td>
<td>4</td>
<td>Engin et al. [67]</td>
<td>2</td>
<td>&lt;=9%</td>
<td>4</td>
</tr>
<tr>
<td>de Chazal and Reilly <a href="ii">292</a></td>
<td>13</td>
<td>76.49%</td>
<td>4</td>
<td>Hosseini et al. [116]</td>
<td>2</td>
<td>&lt;=9%</td>
<td>6</td>
</tr>
<tr>
<td>Christov and Bortolan [16]</td>
<td>12</td>
<td>91.78%</td>
<td>2</td>
<td>Owis et al. [120]</td>
<td>2</td>
<td>&lt;=9%</td>
<td>5</td>
</tr>
<tr>
<td>Belgacem et al. [56]</td>
<td>12</td>
<td>85.19%</td>
<td>2</td>
<td>Engin [293]</td>
<td>9</td>
<td>&lt;=9%</td>
<td>4</td>
</tr>
<tr>
<td>Bortolan et al. <a href="ii">66</a></td>
<td>11</td>
<td>89.20%</td>
<td>2</td>
<td>Özbay et al. [229]</td>
<td>-1</td>
<td>&lt;=9%</td>
<td>10</td>
</tr>
<tr>
<td>Simon and Enwaran [52]</td>
<td>11</td>
<td>&lt;=9%</td>
<td>5</td>
<td>Yang and Liao [101]</td>
<td>-1</td>
<td>&lt;=9%</td>
<td>7</td>
</tr>
<tr>
<td>LLamendo and Martínez [20]</td>
<td>11</td>
<td>53.11%</td>
<td>4</td>
<td>Zhang and Zhang [23]</td>
<td>-1</td>
<td>&lt;=9%</td>
<td>4</td>
</tr>
<tr>
<td>de Chazal et al. [38]</td>
<td>11</td>
<td>54.22%</td>
<td>5</td>
<td>Özdemir and Danisman [177]</td>
<td>-3</td>
<td>&lt;=9%</td>
<td>3</td>
</tr>
<tr>
<td>Yu and Chen [291]</td>
<td>10</td>
<td>99.58%</td>
<td>6</td>
<td>Patra et al. [109]</td>
<td>-5</td>
<td>&lt;=9%</td>
<td>6</td>
</tr>
<tr>
<td>Niwas et al. [71]</td>
<td>10</td>
<td>&lt;=9%</td>
<td>10</td>
<td>Tezel and Özbay [228]</td>
<td>-7</td>
<td>&lt;=9%</td>
<td>10</td>
</tr>
<tr>
<td>Njadadholi et al. [123]</td>
<td>10</td>
<td>&lt;=9%</td>
<td>5</td>
<td>Moeavenian and Khorrari [294]</td>
<td>-8</td>
<td>&lt;=9%</td>
<td>7</td>
</tr>
<tr>
<td>Ham and Han [37]</td>
<td>9</td>
<td>98.56%</td>
<td>2</td>
<td>Khorrari and Moavenian [295]</td>
<td>-8</td>
<td>&lt;=9%</td>
<td>5</td>
</tr>
</tbody>
</table>

Table D.1: Summary of all 72 reviewed publications towards ECG classification sorted with respect to the $\rho$-Score in descending order. For each publication, author, $\rho$-Score (reliability), jx-Index (performance) and number of classes under investigation C are given.
This appendix is intended to transparently show how the $\rho$-Scores were calculated for each publication that was included into the review of section 2.4.

### Song et al. [204]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>28/32</td>
<td>28/32</td>
<td>28/32</td>
<td>28/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Tsipouras et al. [34]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>26/32</td>
<td>26/32</td>
<td>26/32</td>
<td>26/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Ince et al. [163]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>&lt;1%</td>
<td>8/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.8$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td>total points</td>
<td>24/32</td>
<td>24/32</td>
<td>24/32</td>
<td>24/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Lagerholm et al. [47](i)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>unsupervised</td>
<td>2/8</td>
<td>unsupervised</td>
<td>2/8</td>
<td>unsupervised</td>
<td>2/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td>total points</td>
<td>22/32</td>
<td>22/32</td>
<td>22/32</td>
<td>22/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Shyu et al. [111]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>unsupervised</td>
<td>2/8</td>
<td>unsupervised</td>
<td>2/8</td>
<td>unsupervised</td>
<td>2/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td>total points</td>
<td>19/32</td>
<td>19/32</td>
<td>19/32</td>
<td>19/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Lai and Chan [85]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>$r_2$</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>$r_3$</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
<td>supervised</td>
<td>2/2</td>
</tr>
<tr>
<td>$r_4$</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
<td>$\frac{1}{2} = 0.9$</td>
<td>6/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td>total points</td>
<td>19/32</td>
<td>19/32</td>
<td>19/32</td>
<td>19/32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Additional Notes

- **a** LDA selected features
- **b** 3 consecutive RR-intervals
- **c** PCA components of wavelet coefficients + RR
- **d** N, SVEB, PVC, I, Q
- **e** N, V, VI, BI
- **f** N, SVEB, PVC, I, Q
- **g** uses both leads
- **h** 10 of 20 records used
- **i** clustering of data, and afterwards decision which cluster accounts for which beat type
- **j** QRS-area and QRS-duration
- **k** one fractal parameter and the cross correlation
- **l** uses both leads
- **m** clustering of data, and afterwards decision which cluster accounts for which beat type
- **n** QRS-area and QRS-duration
- **o** one fractal parameter and the cross correlation
- **p** uses both leads
- **q** clustering of data, and afterwards decision which cluster accounts for which beat type
- **r** QRS-area and QRS-duration
- **s** one fractal parameter and the cross correlation
- **t** uses both leads
- **u** clustering of data, and afterwards decision which cluster accounts for which beat type
- **v** QRS-area and QRS-duration
- **w** one fractal parameter and the cross correlation
- **x** uses both leads
- **y** clustering of data, and afterwards decision which cluster accounts for which beat type
- **z** QRS-area and QRS-duration
- **aa** one fractal parameter and the cross correlation
- **bb** uses both leads
- **cc** clustering of data, and afterwards decision which cluster accounts for which beat type
- **dd** QRS-area and QRS-duration
- **ee** one fractal parameter and the cross correlation

---

Appendix E

Detailed scoring of reviewed publications
### Appendix E Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Rodriguez et al. [5]</th>
<th>Lagerholm et al. [47]</th>
<th>Jekova et al. [64]</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
</tr>
<tr>
<td>selfunc</td>
<td>unsupervised 2/2</td>
<td>unsupervised 2/2</td>
<td>random 2/2</td>
</tr>
<tr>
<td>rtt</td>
<td>66.7% 0.94</td>
<td>1.1 4/6</td>
<td>5.2 0/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes 3/5</td>
<td>acc, T 1/5</td>
<td></td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No 0/2</td>
<td>No 0/2</td>
<td>No 0/2</td>
</tr>
<tr>
<td>total points</td>
<td>18/32</td>
<td>18/32</td>
<td>18/32</td>
</tr>
<tr>
<td>classes</td>
<td>13 5</td>
<td>13 5</td>
<td>5 5</td>
</tr>
<tr>
<td>acc</td>
<td>96.7%</td>
<td>96.43%</td>
<td>96.31%</td>
</tr>
<tr>
<td>T</td>
<td>86.97%</td>
<td>81.45%</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>99.61%</td>
<td>96.34%</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>99.57%</td>
<td>99.86%</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>99.37%</td>
<td>97.14%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Jiang and Kong [60]</th>
<th>Prasad and Sahambi [17]</th>
<th>Chang et al. [223]</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd</td>
<td>100% 0.94</td>
<td>1.1 4/6</td>
<td></td>
</tr>
<tr>
<td>selfunc</td>
<td>random 2/2</td>
<td>supervised 1/2</td>
<td></td>
</tr>
<tr>
<td>rtt</td>
<td>50.7% 6/6</td>
<td>4/6</td>
<td></td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes 3/5</td>
<td>acc, T 1/5</td>
<td></td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No 0/2</td>
<td>No 0/2</td>
<td>No 0/2</td>
</tr>
<tr>
<td>total points</td>
<td>18/32</td>
<td>17/32</td>
<td>16/32</td>
</tr>
<tr>
<td>classes</td>
<td>13 5</td>
<td>13 5</td>
<td>10 5</td>
</tr>
<tr>
<td>acc</td>
<td>94.49%</td>
<td>96.77%</td>
<td>99.71%</td>
</tr>
<tr>
<td>T</td>
<td>86.62%</td>
<td>99.51%</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>95.30%</td>
<td>--%</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>97.47%</td>
<td>--%</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>78.73%</td>
<td>--%</td>
<td></td>
</tr>
</tbody>
</table>

---

a: uses both leads  b: clustering of data, and afterwards decision which cluster accounts for which beat type  c: only 91 QRS-complexes for training and the rest in testing  d: QRS-ECG/PPwave measures  e: if only the 13 classes are used, that have more than 50 beats in the dataset, no NPREM, no AESC, no Q  f: N, LBRR, BBRR, APC, AAP, NPREM, VPC, VF, VESC, N, VF, PS, FN, P, NPREM, NESC, NVPREM(zero items in set)  g: N, LBRR, BBRR, APC, AAP, NPREM, VPC, VF, VESC, N, VF, PS, FN, P, NPREM, NESC, NVPREM  h: the global set G is the hardest achievable task in current ECG classification

---

xvi
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>r_d</td>
<td>84.00%</td>
<td>2/4</td>
<td>100%</td>
<td>4/4</td>
<td>27.8%</td>
<td>1/4</td>
</tr>
<tr>
<td>r_p</td>
<td>&lt; 1%</td>
<td>8/8</td>
<td>leave one out</td>
<td>4/8</td>
<td>1.9%</td>
<td>8/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>fairly random</td>
<td>1/2</td>
<td>leave one out</td>
<td>2/2</td>
<td>supervised</td>
<td>-1/2</td>
</tr>
<tr>
<td>r_T</td>
<td>$\frac{2}{2} = 14.3^b$</td>
<td>-2/6</td>
<td>$\frac{2}{2} = 11.5^c$</td>
<td>-2/6</td>
<td>$\frac{2}{2} = 4.3^d$</td>
<td>0/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>jk-calculable</td>
<td>2/5</td>
<td>rec.</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>creval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
<td>Yes</td>
<td>2/2</td>
</tr>
<tr>
<td>total points</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
</tr>
<tr>
<td>classes</td>
<td>3 $^f$</td>
<td>2 $^f$</td>
<td>3 $^f$</td>
<td>2 $^f$</td>
<td>3 $^f$</td>
<td>2 $^f$</td>
</tr>
<tr>
<td>acc</td>
<td>95.16%</td>
<td>97.08%</td>
<td>97.14%</td>
<td>94.23%</td>
<td>94.06%</td>
<td>93.20%</td>
</tr>
<tr>
<td>T</td>
<td>98.23%</td>
<td>95.30%</td>
<td>94.23%</td>
<td>90.01%</td>
<td>94.06%</td>
<td>93.20%</td>
</tr>
<tr>
<td>S</td>
<td>~ 0%</td>
<td>84.33%</td>
<td>~ 0%</td>
<td>84.33%</td>
<td>~ 0%</td>
<td>84.33%</td>
</tr>
<tr>
<td>P</td>
<td>95.20%</td>
<td>77.85%</td>
<td>97.09%</td>
<td>93.20%</td>
<td>97.09%</td>
<td>93.20%</td>
</tr>
<tr>
<td>N</td>
<td>~ 0%</td>
<td>97.09%</td>
<td>93.20%</td>
<td>93.20%</td>
<td>93.20%</td>
<td>93.20%</td>
</tr>
</tbody>
</table>

$^a$ 40 records  
$^b$ 42 wavelet coefficients + RR  
$^c$ OAS-morphology parameters  
$^d$ morpho and time-interval params  
$^e$ unknown or other beats are grouped as normal beats  
$^f$ N, VPC, Other

---

### de Chazal and Reilly [292](i)  
### Bortolan et al. [66](c)  
### Jiang and Kong [60](i)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>r_d</td>
<td>100%</td>
<td>4/4</td>
<td>98%</td>
<td>3/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>r_p</td>
<td>50.7%</td>
<td>0/8</td>
<td>&lt;1%</td>
<td>8/8</td>
<td>50.7%</td>
<td>0/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random$^d$</td>
<td>2/2</td>
<td>random</td>
<td>2/2</td>
<td>random$^d$</td>
<td>2/2</td>
</tr>
<tr>
<td>r_T</td>
<td>$\frac{2}{2} = 3^d$</td>
<td>2/6</td>
<td>$\frac{2}{2} = 13^d$</td>
<td>2/6</td>
<td>$\frac{2}{2} = 1.4^d$</td>
<td>4/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>creval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
<td>16/32</td>
</tr>
<tr>
<td>classes</td>
<td>2 $^f$</td>
<td>2 $^f$</td>
<td>2 $^f$</td>
<td>2 $^f$</td>
<td>2 $^f$</td>
<td>2 $^f$</td>
</tr>
<tr>
<td>acc</td>
<td>93.89%</td>
<td>82.02%</td>
<td>94.51%</td>
<td>79.02%</td>
<td>94.51%</td>
<td>79.02%</td>
</tr>
<tr>
<td>T</td>
<td>70.06%</td>
<td>81.41%</td>
<td>67.93%</td>
<td>77.93%</td>
<td>67.93%</td>
<td>77.93%</td>
</tr>
<tr>
<td>S</td>
<td>97.88%</td>
<td>81.41%</td>
<td>94.14%</td>
<td>81.41%</td>
<td>94.14%</td>
<td>81.41%</td>
</tr>
<tr>
<td>P</td>
<td>53.95%</td>
<td>60.41%</td>
<td>84.53%</td>
<td>60.41%</td>
<td>84.53%</td>
<td>60.41%</td>
</tr>
<tr>
<td>N</td>
<td>93.31%</td>
<td>60.41%</td>
<td>96.84%</td>
<td>60.41%</td>
<td>96.84%</td>
<td>60.41%</td>
</tr>
<tr>
<td>x</td>
<td>74.92%</td>
<td>28.20%</td>
<td>78.78%</td>
<td>28.20%</td>
<td>78.78%</td>
<td>28.20%</td>
</tr>
</tbody>
</table>

$^a$ only paced beat records excluded  
$^b$ only paced beat records excluded  
$^c$ global set  
$^d$ inter-patient scheme with randomized records for test or training set  
$^e$ inter-patient scheme with randomized records for test or training set  
$^f$ ECG compressed to 15 DCT components  
$^g$ 23 morphologic and 3 vectorcardiogram features  
$^h$ 7 input features in neural network coming from hermite coefficients  
$^i$ without unknown beat  
$^j$ N, S, V, F, Q  
$^k$ N, VPC  
$^l$ N, S, VQV

---

xvii
### Appendix E: Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT, BIH 3/3</td>
<td>MIT, AHA 3/3</td>
<td>MIT-BIH 3/3</td>
</tr>
<tr>
<td>rd</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
</tr>
<tr>
<td>selfunc</td>
<td>random 2/2</td>
<td>random 2/2</td>
<td>randomd 2/2</td>
</tr>
<tr>
<td>rj</td>
<td>$\frac{2}{3} = 2^2$</td>
<td>$\frac{2}{3} = 2.5^2$</td>
<td>$\frac{2}{3} = 2.3^2$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>No -5/5</td>
<td>No -5/5</td>
<td>Yes 3/5</td>
</tr>
<tr>
<td>crosval-db</td>
<td>No 0/2</td>
<td>Yes 2/2</td>
<td>in trainingi 1/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No 0/2</td>
<td>No 0/2</td>
<td>No 0/2</td>
</tr>
<tr>
<td>total points</td>
<td>16/32</td>
<td>16/32</td>
<td>15/30</td>
</tr>
<tr>
<td>acc</td>
<td>93.55% 3/3</td>
<td>84.39% 3/8</td>
<td>90.74% 3/3</td>
</tr>
<tr>
<td>T</td>
<td>90.74% 3/3</td>
<td>93.91% 3/3</td>
<td>89.22% 3/3</td>
</tr>
<tr>
<td>S</td>
<td>93.26% 3/3</td>
<td>84.39% 3/3</td>
<td>94.86% 3/3</td>
</tr>
<tr>
<td>N</td>
<td>93.55% 3/3</td>
<td>93.68% 3/3</td>
<td>95.83% 3/3</td>
</tr>
<tr>
<td>κ</td>
<td>70.35% 3/3</td>
<td>70.35% 3/3</td>
<td>74.81% 3/3</td>
</tr>
</tbody>
</table>

### Notes:
- $^a$ only paced beat records excluded
- $^b$ only paced beat records excluded
- $^c$ assuming one measures the results of the long-term
- $^d$ inter-patient scheme with randomized records for test or training
- $^e$ 30 PCA components
- $^f$ five principal components
- $^g$ multilead wavelet and vectorcardiogram and morphology parameters after feature selection
- $^h$ only PVC sensitivities were given, no information about normal heartbeats and specificity
- $^i$ AHA and MIT Longterm
- $^j$ MIT-BIH-SUP
- $^k$ N, VPC
- $^l$ N, V
- $^m$ N, S, V
- $^n$ no confusion matrix

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Llamendo and Martínez <a href="ii">40</a></th>
<th>Llamendo and Martínez <a href="i">40</a></th>
<th>Yeap [35]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT, BIH 3/3</td>
<td>MIT, BIH 3/3</td>
<td>MIT, BIH 3/3</td>
</tr>
<tr>
<td>rd</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
<td>100% 4/4</td>
</tr>
<tr>
<td>rs</td>
<td>46.6% 2/8</td>
<td>46.6% 2/8</td>
<td>33% 3/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised -1/2</td>
<td>supervised -1/2</td>
<td>supervised -1/2</td>
</tr>
<tr>
<td>rj</td>
<td>$\frac{2}{5} = 2^{2.0}$</td>
<td>$\frac{2}{5} = 2^{2.6}$</td>
<td>$\frac{2}{5} = 2.5^2$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes 3/5</td>
<td>Yes 3/5</td>
<td>Yes 3/5</td>
</tr>
<tr>
<td>crosval-db</td>
<td>Yes 2/2</td>
<td>Yes 2/2</td>
<td>No 0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No 0/2</td>
<td>No 0/2</td>
<td>No 0/2</td>
</tr>
<tr>
<td>total points</td>
<td>15/32</td>
<td>15/32</td>
<td>14/32</td>
</tr>
<tr>
<td>acc</td>
<td>93.62% 3/3</td>
<td>84.86% 3/3</td>
<td>94.86% 3/3</td>
</tr>
<tr>
<td>T</td>
<td>84.39% 3/3</td>
<td>82.75% 3/3</td>
<td>82.75% 3/3</td>
</tr>
<tr>
<td>S</td>
<td>93.91% 3/3</td>
<td>93.68% 3/3</td>
<td>93.68% 3/3</td>
</tr>
<tr>
<td>P</td>
<td>74.81% 3/3</td>
<td>87.24% 3/3</td>
<td>87.24% 3/3</td>
</tr>
<tr>
<td>N</td>
<td>88.53% 3/3</td>
<td>87.24% 3/3</td>
<td>87.24% 3/3</td>
</tr>
<tr>
<td>κ</td>
<td>70.35% 3/3</td>
<td>69.63% 3/3</td>
<td>69.63% 3/3</td>
</tr>
</tbody>
</table>

### Notes:
- $^a$ unknown, $r_d$ in favor of supervised feature selection
- $^b$ DWT parameters and RR-intervals
- $^c$ DWT parameters and RR-intervals
- $^d$ QRS-morphology
- $^e$ cross-validation results only for three classes
- $^f$ N, V, S
- $^g$ results given only 3 classes without the fusion beat
- $^h$ N, V, S, F
- $^i$ N, Ectopic

xviii
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>r_2</td>
<td>100%(^c)</td>
<td>4/4</td>
<td>100%(^c)</td>
<td>4/4</td>
<td>100%(^c)</td>
<td>4/4</td>
</tr>
<tr>
<td>r_3</td>
<td>50.7%(^d)</td>
<td>6/8</td>
<td>50.7%(^d)</td>
<td>6/8</td>
<td>60%(^d)</td>
<td>0/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>random(^d)</td>
<td>2/2</td>
<td>random(^d)</td>
<td>2/2</td>
<td>random(^d)</td>
<td>2/2</td>
</tr>
<tr>
<td>r_T</td>
<td>0.50%(^e)</td>
<td>2/6</td>
<td>3.75%(^e)</td>
<td>2/6</td>
<td>13%(^e)</td>
<td>2/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>2</td>
<td>No</td>
<td>0/2</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>total points</td>
<td>14/32</td>
<td></td>
<td>13/32(^b)</td>
<td></td>
<td>12/32</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>classes</th>
<th>4(^e)</th>
<th></th>
<th>4(^e)</th>
<th></th>
<th>2(^e)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>acc</td>
<td>86.99%(^d)</td>
<td></td>
<td>94.04%(^d)</td>
<td></td>
<td>98.57%(^d)</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>80.18%(^d)</td>
<td></td>
<td>87.65%(^d)</td>
<td></td>
<td>99.11%(^d)</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>95.71%(^d)</td>
<td></td>
<td>97.41%(^d)</td>
<td></td>
<td>99.11%(^d)</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>56.28%(^d)</td>
<td></td>
<td>67.45%(^d)</td>
<td></td>
<td>90.56%(^d)</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>87.68%(^d)</td>
<td></td>
<td>91.79%(^d)</td>
<td></td>
<td>90.56%(^d)</td>
<td></td>
</tr>
<tr>
<td>κ</td>
<td>59.92%(^d)</td>
<td></td>
<td>75.42%(^d)</td>
<td></td>
<td>88.72%(^d)</td>
<td></td>
</tr>
<tr>
<td>acc</td>
<td>92.77%(^d)</td>
<td></td>
<td>98.15%(^d)</td>
<td></td>
<td>97.09%(^d)</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>91.96%(^d)</td>
<td></td>
<td>97.06%(^d)</td>
<td></td>
<td>96.04%(^d)</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>91.96%(^d)</td>
<td></td>
<td>97.06%(^d)</td>
<td></td>
<td>96.04%(^d)</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>88.52%(^d)</td>
<td></td>
<td>89.11%(^d)</td>
<td></td>
<td>89.11%(^d)</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>88.52%(^d)</td>
<td></td>
<td>89.11%(^d)</td>
<td></td>
<td>89.11%(^d)</td>
<td></td>
</tr>
<tr>
<td>κ</td>
<td>80.14%(^d)</td>
<td></td>
<td>85.31%(^d)</td>
<td></td>
<td>85.31%(^d)</td>
<td></td>
</tr>
</tbody>
</table>
Appendix E: Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Llamendo and Martínez [20]</th>
<th>de Chazal et al. [38]</th>
<th>Yu and Chen [291]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH 3/3</td>
<td>MIT-BIH 3/3</td>
<td>MIT-BIH 3/3</td>
</tr>
<tr>
<td>( r_d )</td>
<td>100%(^a)</td>
<td>100%</td>
<td>11.2%</td>
</tr>
<tr>
<td>( r_p )</td>
<td>50.7%</td>
<td>46.6%</td>
<td>50%</td>
</tr>
<tr>
<td>selfunc</td>
<td>random(^b)</td>
<td>supervised-1/2</td>
<td>supervised(^d)-1/2</td>
</tr>
<tr>
<td>( r_f )</td>
<td>( \frac{2}{3} = 0.67 )</td>
<td>( \frac{2}{3} = 10.4' )</td>
<td>( \frac{2}{3} = 1.8' )</td>
</tr>
<tr>
<td>conffmatix</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>11/32</td>
<td>11/32</td>
<td>10/32</td>
</tr>
<tr>
<td>classes</td>
<td>4(^a)</td>
<td>5(^a)</td>
<td>6(^a)</td>
</tr>
<tr>
<td>acc</td>
<td>79.75%</td>
<td>85.88%</td>
<td>99.65%</td>
</tr>
<tr>
<td>T</td>
<td>79.65%</td>
<td>45.57%</td>
<td>99.61%</td>
</tr>
<tr>
<td>S</td>
<td>93.97%</td>
<td>88.85%</td>
<td>99.93%</td>
</tr>
<tr>
<td>P</td>
<td>48.56%</td>
<td>66.00%</td>
<td>99.60%</td>
</tr>
<tr>
<td>N</td>
<td>83.64%</td>
<td>96.05%</td>
<td>99.93%</td>
</tr>
<tr>
<td>( \kappa )</td>
<td>42.11%</td>
<td>52.65%</td>
<td>99.55%</td>
</tr>
<tr>
<td>acc</td>
<td>assuming that only 10406 beats were used for testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>RR-interval, QRS-duration, spectral entropy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Gaussian mixture models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2 linear prediction coefficients, mean squared QRS-value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>accuracies are doubtable, how is it possible to obtain 98.9%accuracy if one tests on 20 beats, accuracy should be 100%/or 95%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kappa</td>
<td>under the assumption the accuracy really meant sensitivity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Niwas et al. [71]</th>
<th>Nejadgholi et al. [123]</th>
<th>Ham and Han [37]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH 3/3</td>
<td>MIT-BIH 3/3</td>
<td>AHA 3/3</td>
</tr>
<tr>
<td>( r_d )</td>
<td>&lt;1%</td>
<td>10.4%</td>
<td>12.1%</td>
</tr>
<tr>
<td>( r_p )</td>
<td>58.1%</td>
<td>6/8</td>
<td>75%</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>random</td>
<td>supervised-1/2</td>
</tr>
<tr>
<td>( r_f )</td>
<td>( \frac{2}{3} = 0.67 )</td>
<td>( \frac{2}{3} = 3.2' )</td>
<td>( \frac{2}{3} = 1.5' )</td>
</tr>
<tr>
<td>conffmatix</td>
<td>arc, ( T )</td>
<td>No-5/5'</td>
<td>rec. 3/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>10/32</td>
<td>10/32</td>
<td>9/32</td>
</tr>
<tr>
<td>classes</td>
<td>10(^a)</td>
<td>5(^a)</td>
<td>1(^a)</td>
</tr>
<tr>
<td>acc</td>
<td>96.49%(^a)</td>
<td>92.5%</td>
<td>99.44%</td>
</tr>
<tr>
<td>T</td>
<td>96.06%(^b)</td>
<td>92.98%(^b)</td>
<td>98.66%</td>
</tr>
<tr>
<td>S</td>
<td>97.12%(^c)</td>
<td>97.12%(^c)</td>
<td>98.66%</td>
</tr>
<tr>
<td>P</td>
<td>( \sim % )</td>
<td>( \sim % )</td>
<td>99.43%</td>
</tr>
<tr>
<td>N</td>
<td>( \sim % )</td>
<td>( \sim % )</td>
<td>99.43%</td>
</tr>
<tr>
<td>kappa</td>
<td>( \sim % )</td>
<td></td>
<td>98.08%</td>
</tr>
</tbody>
</table>

\(^a\) assuming that only 10406 beats were used for testing  
\(^b\) RR-interval, QRS-duration, spectral entropy  
\(^c\) Gaussian mixture models  
\(^d\) 2 linear prediction coefficients, mean squared QRS-value  
\(^e\) accuracies are doubtable, how is it possible to obtain 98.9%accuracy if one tests on 20 beats, accuracy should be 100%/or 95%  
\(^f\) sensitivity and specificity are only given in a plot not as raw numbers  
\(^g\) N, LBBB, RBBB, APC, VPC, VF  
\(^h\) N, LBBB, RBBB, APC, AE, VPC, AE, VF, SS, IVF  
\(^i\) N, LBBB, RBBB, VPC, VF  
\(^j\) under the assumption the accuracy really meant sensitivity  
\(^k\) under the assumption the accuracy really meant sensitivity  

XX
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT</td>
<td>3/3</td>
<td>MIT</td>
<td>3/3</td>
</tr>
<tr>
<td>r_d</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
<td>100%</td>
<td>4/4</td>
</tr>
<tr>
<td>r_s</td>
<td>50.7%</td>
<td>0/6</td>
<td>50.7%</td>
<td>0/6</td>
<td>97.9%</td>
<td>0/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>randomf</td>
<td>2/2</td>
<td>randomf</td>
<td>2/2</td>
<td>randomf</td>
<td>2/2</td>
</tr>
<tr>
<td>r_f</td>
<td>$\frac{2}{\pi} = 8.3^d$</td>
<td>-1/6</td>
<td>$\frac{2}{\pi} = 7.5^a$</td>
<td>-1/6</td>
<td>$\frac{2}{\pi} = 18.68^d$</td>
<td>-2/6</td>
</tr>
<tr>
<td>confmatrix</td>
<td>acc, $T$</td>
<td>1/5</td>
<td>acc, $T$</td>
<td>1/5</td>
<td>acc, $T$</td>
<td>1/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>9/32</td>
<td>9/32</td>
<td>8/32</td>
<td>8/32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>4 $^c$</td>
<td>4 $^c$</td>
<td>3 $^c$</td>
<td>3 $^c$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acc</td>
<td>95.56%</td>
<td>76.43%</td>
<td>99.1%</td>
<td>89.1%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>76.16%</td>
<td>83.00%</td>
<td>81.0%</td>
<td>81.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>\textless -4%</td>
<td>93.88%</td>
<td>\textless -4%</td>
<td>93.88%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>\textless -4%</td>
<td>39.68%</td>
<td>\textless -4%</td>
<td>39.68%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\kappa$</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td>\textless -4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>crossval-db</td>
<td>No $^d$</td>
<td>-5/5</td>
<td>Yes</td>
<td>3/5</td>
<td>Yes</td>
<td>3/5</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>6/32</td>
<td>7/32</td>
<td>7/32</td>
<td>7/32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>2 $^e$</td>
<td>2 $^e$</td>
<td>3 $^e$</td>
<td>3 $^e$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$^a$ only paced beat records excluded  
$^b$ only paced beat records excluded  
$^c$ 48-fold cross-validation  
$^d$ five folds train, one fold test  
$^e$ inter-patient scheme with randomized records for test or training set  
$^f$ leaving one file out  
$^g$ 10 HOS, 20 HBF and 3 RR-intervals  
$^h$ 6 RR + 24 QRS-morphology features  
$^i$ QRS-features + RR-interval features  
$^j$ N, V, F  
$^k$ N, PVC, S, F  
$^l$ N, V  

$^a$ no clear record documentation  
$^b$ the proprietary data consists of approx. 800,000 beats from 8 records for test and approx. 500,000 for training (train and cross-validation)  
$^c$ filtered ECG as MLP input  
$^d$ template matching correlation coefficient  
$^e$ no numbers of used normal and VPC beats are given  
$^f$ N, VPC  
$^g$ N, V  
$^h$ N, N, APC
### Appendix E: Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Coast et al. [222]</th>
<th>Abibullaev and Seo [105]</th>
<th>Ge et al. [90]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>AHA</td>
<td>MIT-BIH</td>
<td>MIT-BIH</td>
</tr>
<tr>
<td>rd</td>
<td>15 %</td>
<td>&lt;1%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>rs</td>
<td>50 %</td>
<td>76%</td>
<td>50%</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>unknown</td>
<td>supervised</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rj</td>
<td>2/3</td>
<td>2/6</td>
<td>2/2</td>
</tr>
<tr>
<td>condmatr</td>
<td>rec.</td>
<td>acc, T</td>
<td>acc, T</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>7/32</td>
<td>7/32</td>
<td>7/32</td>
</tr>
<tr>
<td>classes</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Achr [61]</th>
<th>Froese et al. [70]</th>
<th>Zhou [230]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>MIT-BIH</td>
<td>MIT-BIH</td>
</tr>
<tr>
<td>rd</td>
<td>1.6 %</td>
<td>9.8%</td>
<td>10.3%</td>
</tr>
<tr>
<td>rs</td>
<td>50 %</td>
<td>13.7%</td>
<td>75%</td>
</tr>
<tr>
<td>selfunc</td>
<td>random</td>
<td>supervised</td>
<td>supervised</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rj</td>
<td>2/3</td>
<td>2/6</td>
<td>2/2</td>
</tr>
<tr>
<td>condmatr</td>
<td>acc, T</td>
<td>acc, T</td>
<td>acc, T</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>7/32</td>
<td>6/32</td>
<td>6/32</td>
</tr>
<tr>
<td>classes</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

| Criteria |  |  |  |
|----------|  |  |  |
| acc      | 98.94% | 97.82% | 96.86% |
| T        | 98.14% | 97.82% | 96.78% |
| P        | 92.75% | 97.75% | 97.75% |
| N        | 92.75% | 92.75% | 92.75% |
| κ        | 90.53% | 90.53% | 90.53% |
|  |  |  |  |
| acc      | 94.4% | 99.16% | 97.03% |
| T        | 94.2% | 99.60% | 98.51% |
| S        | 89.2% | -      | -      |
| P        | -    | -      | -      |
| N        | -    | -      | -      |
| κ        | -    | -      | -      |

\[a\] use 12 of 80 recordings  \[b\] 250 data samples from normal and abnormal  \[c\] probability of six hidden markov models  \[d\] positive and negative wavelet amplitudes, wavelet wave duration  \[e\] AR coefficients  \[f\] cross-validation only for QRS-detection  \[g\] N, V  \[h\] normal, abnormal  \[i\] N, VPC, APC, SVT, VT, VF  \[j\] calculated from paper data  \[k\] used substitution rates and rejection rates
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
<td>3/3</td>
</tr>
<tr>
<td>( r_d )</td>
<td>(&lt;1%^a)</td>
<td>2/4</td>
<td>1.0%</td>
<td>2/4</td>
<td>5.6%b</td>
<td>0/4</td>
</tr>
<tr>
<td>( r_m )</td>
<td>50%c</td>
<td>0/8</td>
<td>50%</td>
<td>0/8</td>
<td>50%</td>
<td>0/8</td>
</tr>
<tr>
<td>selfunc</td>
<td>unknown</td>
<td>-1/2</td>
<td>supervisedd</td>
<td>-1/2</td>
<td>supervised</td>
<td>-1/2</td>
</tr>
<tr>
<td>( r_f )</td>
<td>( \frac{2}{5} )</td>
<td>2/6</td>
<td>( \frac{2}{5} )</td>
<td>1.5f</td>
<td>( \frac{2}{5} )</td>
<td>2.5f</td>
</tr>
<tr>
<td>confmatrix</td>
<td>yes</td>
<td>3/5</td>
<td>acc, T</td>
<td>1/5</td>
<td>acc, T</td>
<td>1/5</td>
</tr>
<tr>
<td>crossval-db</td>
<td>no</td>
<td>0/2</td>
<td>no</td>
<td>0/2</td>
<td>no</td>
<td>0/2</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>no</td>
<td>0/2</td>
<td>no</td>
<td>0/2</td>
<td>no</td>
<td>0/2</td>
</tr>
<tr>
<td>total points</td>
<td>6/32</td>
<td>5/32</td>
<td>5/32</td>
<td>5/32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>10g</td>
<td>10h</td>
<td>7j</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acc</td>
<td>100.00%</td>
<td>97.00%</td>
<td>97.28%</td>
<td>97.78%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>100.00%</td>
<td>99.66%</td>
<td>96.52%</td>
<td>96.52%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>100.00%</td>
<td>---%</td>
<td>---%</td>
<td>---%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>100.00%</td>
<td>---%</td>
<td>---%</td>
<td>---%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>100.00%</td>
<td>---%</td>
<td>---%</td>
<td>---%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \kappa )</td>
<td>100.00%</td>
<td>---%</td>
<td>---%</td>
<td>---%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Notes:**
- a: author uses 4 beats per class as test
- b: supervised selection of records and beats used
- c: 106 beans from MIT3342 proprietary
- d: special records left out
- e: direct input of ECG samples
- f: wavelet coefficients after feature selection
- g: variety of wavelet features
- h: N, Brady, VT, Tachocardiac, APC, fM, RBBB, LBBB, AFlut, APC
- i: N, LBBB, RBBB, S, L, F, f, a, E, P, p, V
- j: N, LBBB, RBBB, APC, VPC, VESC, VFW
- k: Snormal = 97.80%
### Appendix E: Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Hu et al. [30]</th>
<th>Yu and Chou [110]</th>
<th>Narny et al. [91]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Score</td>
<td>Value</td>
</tr>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
</tr>
<tr>
<td>( r_s )</td>
<td>70.6%</td>
<td>0/4</td>
<td>50%</td>
</tr>
<tr>
<td>( r_m )</td>
<td>50%</td>
<td>0/6</td>
<td>50%</td>
</tr>
<tr>
<td>selffunc</td>
<td>supervised(^d)</td>
<td>1/2</td>
<td>supervised(^d)</td>
</tr>
<tr>
<td>( r_f )</td>
<td>( \frac{4}{7} = 0.57 )</td>
<td>0/6</td>
<td>( \frac{5}{8} = 0.625 )</td>
</tr>
<tr>
<td>confmatrix</td>
<td>No(^i)</td>
<td>0/2</td>
<td>No(^i)</td>
</tr>
<tr>
<td>crossover-db</td>
<td>Yes</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>Yes</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>classes</td>
<td>4(^c)</td>
<td>6(^b)</td>
<td>6(^b)</td>
</tr>
<tr>
<td>acc</td>
<td>-</td>
<td>-</td>
<td>99.51%</td>
</tr>
<tr>
<td>T</td>
<td>99.9%</td>
<td>-</td>
<td>99.9%</td>
</tr>
<tr>
<td>S</td>
<td>97.01%</td>
<td>-</td>
<td>97.01%</td>
</tr>
<tr>
<td>P</td>
<td>-</td>
<td>-</td>
<td>99.40%</td>
</tr>
<tr>
<td>N</td>
<td>-</td>
<td>-</td>
<td>99.40%</td>
</tr>
<tr>
<td>( \kappa )</td>
<td>-</td>
<td>-</td>
<td>97.00%</td>
</tr>
</tbody>
</table>

\(^{a}\) deleted 11 records because they do not contain PVCs, erroneous because this artificially increases specificity  
\(^{b}\) GE expert trained on all 100-124 minus excluded and LE is not divided by 12  
\(^{c}\) special files from the db were selected for training and test set  
\(^{d}\) specific files were selected for analysis, a major drawback for specificity calculation, because only from 6 files the normal beats were selected, the major argument against this is that the ICA maybe just detects the different patients and not the morphologies itself  
\(^{e}\) no real information provided about used records  
\(^{f}\) no transformed, QRS-likelihood, RR-intervals  
\(^{g}\) direct input of ECG samples  
\(^{h}\) no exact information, assumed all ICA components, RR-interval and Power spectra were used  
\(^{i}\) Acc + T calculated only for PVC  
\(^{j}\) N, LBBB, RBBB, SVPREM, NESC, AESC = class1, VPC, APC, VPREM, RestT = class2, P70r=class3, other=class4  
\(^{k}\) N, LBBB, RBBB, VPC, APC, vP  
\(^{l}\) N, VPC, APC, V, VS, VS  

### Appendix E: Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Score</td>
<td>Value</td>
</tr>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
</tr>
<tr>
<td>( r_s )</td>
<td>5.6%</td>
<td>0/4</td>
<td>1.5%</td>
</tr>
<tr>
<td>( r_m )</td>
<td>50%</td>
<td>0/4</td>
<td>50%</td>
</tr>
<tr>
<td>selffunc</td>
<td>supervised(^d)</td>
<td>1/2</td>
<td>supervised(^d)</td>
</tr>
<tr>
<td>( r_f )</td>
<td>( \frac{4}{7} = 0.57 )</td>
<td>0/6</td>
<td>( \frac{5}{8} = 0.625 )</td>
</tr>
<tr>
<td>confmatrix</td>
<td>acc, T</td>
<td>1/5</td>
<td>acc, T</td>
</tr>
<tr>
<td>crossover-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>classes</td>
<td>4(^c)</td>
<td>10(^b)</td>
<td>4(^c)</td>
</tr>
<tr>
<td>acc</td>
<td>96.94%</td>
<td>95.7%</td>
<td>98.23%</td>
</tr>
<tr>
<td>T</td>
<td>96.95%</td>
<td>95.7%</td>
<td>98.23%</td>
</tr>
<tr>
<td>S</td>
<td>-</td>
<td>-</td>
<td>95.5%</td>
</tr>
<tr>
<td>P</td>
<td>-</td>
<td>-</td>
<td>93.18%</td>
</tr>
<tr>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \kappa )</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^{a}\) supervised selection of records and beat used  
\(^{b}\) special records left out  
\(^{c}\) variety of wavelet features after feature reduction (\( F = 30 \) to start features)  
\(^{d}\) no real information provided about used records  
\(^{e}\) choosing DCT feature size after feature reduction  
\(^{f}\) N, V, VS, VP, APC, vP  
\(^{g}\) N, LBBB, VPC, non-PVC wavelet  
\(^{h}\) \( T_{\text{normal}} = 97.78 \%  

xxiv
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Engin et al. [67]</th>
<th>Hosseini et al. [116]</th>
<th>Oweis et al. [120]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Score</td>
<td>Value</td>
</tr>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH</td>
</tr>
<tr>
<td>rs</td>
<td>7.6%</td>
<td>1/4</td>
<td>1.8%</td>
</tr>
<tr>
<td>rpm</td>
<td>50%</td>
<td>0/8</td>
<td>50%</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>1/-2</td>
<td>unknown</td>
</tr>
<tr>
<td>rj</td>
<td>$\frac{4}{3}$</td>
<td>4/6</td>
<td>$\frac{4.2}{2}$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>No</td>
<td>5/-5</td>
<td>acc, T</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>2/32</td>
<td></td>
<td>2/32</td>
</tr>
<tr>
<td>classes</td>
<td>4 $^a$</td>
<td></td>
<td>6 $^c$</td>
</tr>
<tr>
<td>acc</td>
<td>92.45%</td>
<td>90.15%</td>
<td>92.52%</td>
</tr>
<tr>
<td>P</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>N</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Engin [293]</th>
<th>Özbay et al. [229]</th>
<th>Yang and Liao [101]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Score</td>
<td>Value</td>
</tr>
<tr>
<td>ecg-db</td>
<td>MIT-BIH</td>
<td>3/3</td>
<td>MIT-BIH + prop $^d$</td>
</tr>
<tr>
<td>rs</td>
<td>1.4%</td>
<td>2/-4</td>
<td>5% $^b$</td>
</tr>
<tr>
<td>rpm</td>
<td>66.67%</td>
<td>0/8</td>
<td>2.0% $^b$</td>
</tr>
<tr>
<td>selfunc</td>
<td>supervised</td>
<td>1/-2</td>
<td>partly rand.</td>
</tr>
<tr>
<td>rj</td>
<td>$\frac{4}{3}$</td>
<td>0/6</td>
<td>$\frac{20}{2}$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>acc, T</td>
<td>1/5</td>
<td>$\frac{12}{2}$</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>0/2</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>1/32</td>
<td></td>
<td>1/32</td>
</tr>
<tr>
<td>classes</td>
<td>4 $^f$</td>
<td></td>
<td>10 $^f$</td>
</tr>
<tr>
<td>acc</td>
<td>96.00%</td>
<td>99.78%</td>
<td>99.16%</td>
</tr>
<tr>
<td>T</td>
<td>96.00%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>S</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>P</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>N</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>~-%</td>
<td></td>
<td>~-%</td>
</tr>
</tbody>
</table>

---

$^a$ even worse records comprise special arrhythmias not available in other records  
$^b$ 4 frames of autoregressive model  
$^c$ direct input of ECG samples + QRS-features  
$^d$ assuming kNN with PCA, according to Table 7  
$^e$ The author do not provide information about the results on 1 of 4 records  
$^f$ class1: N, VPC = APC, LBBB, MBBB  
$^g$ prop, MIT-BIH 3, MIT-BIH 4  
$^h$ N, V couplet, VCP, V  
$^i$ input of ECG samples  
$^j$ for 4 of 5 classes one of 4 records account for more than 95% of all class samples, i.e. if one can discriminate between the 4 records with a certain feature one will as well classify each beat correctly. Even though the authors pool non-normal beats together, this approach will result in non-generalizable results  
$^k$ MIT not NET_BST, the NET_BST approach is flawed, using stage 1 of NETS will lead to different beat being in the Gp classifier therefore results of the NETS stage two will be different from the results gained when NETS stage two was combined with NETS stage one. Therefore, accuracies are not simply combinable as the author proposes. A good hint is that no results were shown for this network architecture  

---

Engin et al. [67] / Oweis et al. [120]
## Appendix E  Detailed scoring of reviewed publications

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Zhang and Zhang [23]</th>
<th>Özdemir and Danisman [177]</th>
<th>Patra et al. [109]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value Score</td>
<td>Value Score</td>
<td>Value Score</td>
</tr>
<tr>
<td>ecg-db</td>
<td>MTT-BIH 3/3</td>
<td>MTT-BIH 3/3</td>
<td>MTT-BIH 3/3</td>
</tr>
<tr>
<td>rₐ</td>
<td>6.4%</td>
<td>4.8%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>rₐ</td>
<td>66.9%</td>
<td>66.6%</td>
<td>50%</td>
</tr>
<tr>
<td>selfunc random</td>
<td>2/2</td>
<td>supervised -1/2</td>
<td>supervised -1/2</td>
</tr>
<tr>
<td>rₐ</td>
<td>$\frac{2}{3}$</td>
<td>$\frac{2}{3}$</td>
<td>$\frac{2}{3}$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>total points</td>
<td>-1/32</td>
<td>-3/32</td>
<td>-5/32</td>
</tr>
<tr>
<td>classes</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>acc</td>
<td>~99.08%</td>
<td>~97.66%</td>
<td>~97.66%</td>
</tr>
<tr>
<td>T</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>S</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>P</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>N</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>κ</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value Score</td>
<td>Value Score</td>
<td>Value Score</td>
</tr>
<tr>
<td>ecg-db</td>
<td>part. proprietary</td>
<td>MTT-BIH 3/3</td>
<td>MTT-BIH 3/3</td>
</tr>
<tr>
<td>rₐ</td>
<td>~&lt;1%</td>
<td>0.6%</td>
<td>0.4%</td>
</tr>
<tr>
<td>rₐ</td>
<td>28.3%</td>
<td>88.9%</td>
<td>83.3%</td>
</tr>
<tr>
<td>selfunc supervised</td>
<td>~1/2</td>
<td>supervised ~1/2</td>
<td>supervised ~1/2</td>
</tr>
<tr>
<td>rₐ</td>
<td>$\frac{2}{3} = 20^d$</td>
<td>$\frac{2}{3} = 47.7^d$</td>
<td>$\frac{2}{3} = 66.8^d$</td>
</tr>
<tr>
<td>confmatrix</td>
<td>No $^e$</td>
<td>No $^g$</td>
<td>No $^g$</td>
</tr>
<tr>
<td>crossval-db</td>
<td>No $^e$</td>
<td>No $^g$</td>
<td>No $^g$</td>
</tr>
<tr>
<td>unknown-beat</td>
<td>No $^e$</td>
<td>No $^g$</td>
<td>No $^g$</td>
</tr>
<tr>
<td>total points</td>
<td>77/30</td>
<td>87/32</td>
<td>87/32</td>
</tr>
<tr>
<td>classes</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>acc</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>T</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>S</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>P</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>N</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
<tr>
<td>κ</td>
<td>~--%</td>
<td>~--%</td>
<td>~--%</td>
</tr>
</tbody>
</table>

---

$^a$ samples randomly chosen, but records were chosen partly disjoint between classes  
$^b$ choosing the setup with 20, instead of 30 principal components, because accuracy increase is too low  
$^c$ MIT and PCA  
$^d$ N, LBBB, VPC  
$^e$ N, LBBB, VPC, APC  
$^f$ 20 PCA components + SVM  
$^g$ MIT-BIH (train) / proprietary (test)  
$^h$ MIT, 260 samples proprietary  
$^i$ 126 training, 268 test  
$^j$ used values for SVM, for MLP $\eta = 89.9$  
$^k$ 10-defined patients not all of study [229]  
$^l$ special files from the db were selected for training and test set  
$^m$ special files from the db were selected for training and test set  
$^n$ variety of wavelet coefficients  
$^o$ 334 ECG points as input for NN  
$^p$ 334 ECG points as input for NN  
$^q$ testing errors as mean difference in NN output, does not give a proper guess of the accuracy  
$^r$ calculation of the mean squared distance between NN outputs is invalid, because classification does not follow a cardinal scale  
$^s$ calculation of the mean squared distance between NN outputs is invalid, because classification does not follow a cardinal scale  
$^t$ N, Brady VT, Sinuater, APC, VF, LBBB, VPC, APC  
$^u$ LBBB, RBBB, N, VPC, APC, FPN, VF  
$^v$ LBBB, RBBB, N, VPC, APC  
$^w$ results are generated in the same fashion as in [229], which is erroneous
Bibliography


Bibliography


xxx


[107] R. Ghongade and A. Ghatol. A robust and reliable ECG pattern classification using QRS mor-


[123] I. Nejadgholi, M.H. Moradi, and F. Abdolali. Using phase space reconstruction for patient inde-


Bibliography


xxxvi


Acknowledgement

At the beginning of a long journey, only two things are certain: the place you are and the place you want to go, even if your destination is hidden behind mountains and valleys. This last chapter is devoted to all the wonderful people that gave me three years full of excitement, pleasure, fun, knowledge and self-improvement.

I'm very grateful for having Prof. Dr. Bernhard Hensel not only as my supervisor but also as my mentor. During the last 6 years he always supported all my projects with trust, creativity and assurance. Without him I would have never come to know the great Russian soul, many magnificent Russian researchers and surely I would have never understood what they wanted to get across. A big thanks for his good faith in my abilities as junior-IT-administrator at the professorship, which even a broken hard disk could not disenchant. I thank him for all his support, great discussions, the memorable daily breakfast talks and the time he had to spent reading this thesis.

Speaking of reading this thesis. My sincerest thanks to Prof. Dr. Bjoern Eskofier for his valuable remarks on the thesis and the time he carved out of his schedule. I'm very thankful for having a great expert on the field of pattern recognition and ECG analysis reading this thesis.

My sincerest thanks go to Dr. Thomas Krämer for having him around the last four years. I will greatly miss his intelligence, fury, scintillating wit and his company during lunch, traffic jams on the highway and countless prolonged walks. I'm very grateful for his open ears and direct and constructive critic with respect to my thesis and professional life. Without his encouragement and vigor I would have spent 3 further years waiting for the results of my calculations. Big thanks for pulling of this coup de main.

I want to gratefully thank Biotronik SE & Co. KG. for the financial support in terms of salary, conference funds and the 4 IBM-X-servers that allowed me to accomplish writing this thesis just in time. A great thanks to Dr. Wolfgang Meyer, Dr. Albrecht Urbaszek und Dr. Gerhard Czygan for supporting my research financially and with helpful recommendations.

A warm thanks to Dr. Alexander Rzany for all his work and endurance holding the IT-systems running and evolving. I hope my assistance helped him a little reducing his workload towards the IT. The last three years opened a whole new world for me and I'm very grateful to having him guiding me through it. I will always remember our night-shifts making the voodoo-magic happen. My sincerest thanks to Dr. Olaf Rottler for all his passion, his support and his knowledge in our IT-projects. I really appreciated working with two great unpretentious, sharp-witted, steady and fun guys.

A big thanks to the whole team of Biotronik Cardiac Biosignals with whom I shared the last four years working on projects improving people's health and who supported me building up the huge biosignal and machine learning framework. Thank you Manuel and Wolfgang for teaching me statistics and Frankonian card games. My sincerest thanks to Daniel for introducing me to the linux-magic world. Now I really embrace the command line and the package-manager. Big thanks to Oleg for great scientific discussions and for always touching sore points. A warm thanks to Sergej who was always helpful and excited when facing mathematical problems a mathematician like him would typically laugh at.
I'm very grateful for the opportunity to work in Biotronik's cardiac rhythm management department and get to know many great minds and engineers who share the same desire for technology and medicine. Above all I want to thank Jens Potschadtke, the good shepherd, for pulling of his sisyphean task building a magnificent device for the long-awaited clinical trial. I'm moreover very thankful to have worked together with Dr. Jens Kirchner. It is always a great joy to see this bright, deep and artistically inclined person, surprising everybody with his dry humor and shrewd wit. Together with Thomas we had a great time learning, experimenting and writing the patent.

In the last five years, during my time at the professorship, many colleagues came and went it is quite hard to keep track of all of them. But I want to mention a few dear colleagues.

Above all I want to thank Philipp for being this marvelous, self-aware and heartful person. I'm very grateful for your openness, your blunt critique both personally and professionally and the great times we had on the badminton-court and at Hiro.

Big thanks to my dearest roommate Werner. We stole many hours from each other with just talking about software, politics, people and exchanging youtube videos. In fact, those hours have never been stolen, they were a wonderful part of my day. Keep up the good work and hopefully we will sometime develop software together.

I want to thank Florian, Hermann, Johannes, Kerstin and Torsten for the great time we had in Russia, during skiing and all other leisure activities. Warm thanks to Uli and Birgit who's laughter, wit and humor I sincerely miss and who kept me running if situations were dire.

Even though, one spends long times at the office, I'm very grateful that the other half of my life is filled with wonderful and inspirational people that I'm honored to call friends.

Above all I want to thank my girlfriend Eva who is constantly surprising and fascinating me. She's the only person who intuitively knows when 'How is your thesis doing?' is an absolute no go and the only person who insists on understanding every aspect of my work even if I have to come up with queer analogies to make it understandable. The last years have been the most amazing of my life and your support, love, courage, humor, patience and power contributed a lot. I'm looking forward to our next adventures. This time with less type-writing.

My warmest and sincerest thanks go to my sister Claudia, who is the strongest and kindest person I've ever met. Thank you for all your heart, friendship, wisdom and endurance during the last 20 years.

Big thanks to Carina for sharing our desire for dance, music and people. Thank you Falk for always pushing buttons and people forward. You are a constant inspiration for all of us. I am very grateful for Irene, the best flatmate imaginable. Thank you so much for feeding me both intellectually, with cookies and vegi-stuff. My warmest thanks to Sarah, Harald, Anastacia and Max for pushing me out of my chamber into the number-free area and for the great time in Nuremberg.

Finally, I want to thank Ms. Chantal dos Santos for proof-reading the thesis. All late changes to the thesis can be easily identified by comma, hyphenation and sentence structure errors. I'm very grateful for your support and patience, I could have never done this without you.
Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

München, 21. März 2013

Christian Rockstroh