

Mehmet Mert Duman

**PERSONALIZED AND ZERO-SHOT  
ELECTROCARDIOGRAM  
ARRHYTHMIA MONITORING  
SYSTEM**

Master of Science Thesis  
Faculty of Information Technology and Communication Sciences (ITC)  
Thesis Examiners: Moncef Gabbouj,  
Serkan Kiranyaz,  
Mehmet Yamac  
December 2024

# Abstract

Mehmet Mert Duman: Personalized and Zero-Shot Electrocardiogram Arrhythmia Monitoring System

Master of Science Thesis

Tampere University

Master's Degree Education in Computing Sciences

Signal Processing and Machine Learning

October 2024

---

Electrocardiograms (ECG) are a non-intrusive and highly accurate tool for monitoring cardiac abnormalities. They are widely used in clinical settings and can be measured on wearable devices such as smartwatches and Holter monitors. Early detection of cardiac arrhythmias through continuous ECG monitoring has the potential to reduce mortality rates and improve lives. This thesis presents a comprehensive and efficient pipeline for personalized and zero-shot ECG arrhythmia detection suitable for low-end wearable devices. Earlier methods for arrhythmia detection require labeled healthy and anomalous heartbeats for training; however, such data is often unavailable, particularly for healthy individuals with no history of cardiac disorders.

This thesis addresses a real-world scenario where a healthy individual requires continuous monitoring for early detection and prevention of potential cardiac issues. First, we propose a lightweight representation error-based arrhythmia detection method using the nullspaces of the learned sparse representation dictionaries. Our lightweight method reduces computational complexity while maintaining detection accuracy. Then, we improve representation error-based detection performance via compact dictionaries learned through single hidden-layer sparse autoencoders. Then, we propose a sparse representation-based domain adaptation technique to learn morphology transformation matrices for the target individual, which are used to adapt the morphology of existing ECG data from other patients to the domain of the target. This enables a zero-shot classifier to be trained for a healthy individual without any anomalous beats. Then, we design an ensemble classifier by combining the lightweight representation error-based classifier with the zero-shot classifier and achieve an accuracy of 98.2% and an F1-Score of 92.8%. Finally, we propose an energy-efficient monitoring system for mobile devices capable of automatically classifying up to 40% of the test samples with minimal computation, making continuous monitoring feasible for wearable sensors.

**Keywords:** Personalized ECG Monitoring, Zero-shot Arrhythmia Detection, Sparse Representation, Dictionary Learning

The originality of this thesis has been checked using the Turnitin Originality Check service.

# Contents

1	Introduction . . . . .	2
1.1	Motivation . . . . .	4
1.2	Contributions . . . . .	4
1.3	Thesis Structure . . . . .	5
2	Background . . . . .	6
2.1	Electrocardiogram . . . . .	6
2.1.1	Salient Characteristics of ECG . . . . .	6
2.1.2	Categories of ECG Arrhythmia . . . . .	7
2.1.3	Personalized and Zero-shot Arrhythmia Detection . . . . .	8
2.2	Sparse Representation-based Classification . . . . .	9
2.2.1	Notations . . . . .	9
2.2.2	Sparse Representation . . . . .	9
2.2.3	Dictionary Learning . . . . .	11
2.2.4	Sparse and Collaborative Representation-based Classification . . . . .	13
2.2.5	Sparse Representation for ECG Classification . . . . .	14
2.3	Methods for Cardiac Arrhythmia Detection . . . . .	15
2.4	Arrhythmia Synthesis for Detection . . . . .	16
2.4.1	GAN-based Synthesizers . . . . .	17
3	Methods . . . . .	19
3.1	Dictionary Learning for the Healthy ECG Space . . . . .	19
3.2	Representation Error-based Arrhythmia Detection . . . . .	20
3.2.1	Nullspace Projection Error . . . . .	22
3.2.2	Least Squares Approximation Error . . . . .	24
3.2.3	Computational Complexity Analysis . . . . .	25
3.3	Joint Embedding and Dictionary Learning via Personalized Autoencoders . . . . .	26
3.3.1	Autoencoder Reconstruction Error . . . . .	28
3.4	Sparse Representation-based Domain Adaptation . . . . .	29
3.5	Personalized Domain-adapted Arrhythmia Detection . . . . .	34
3.6	Ensemble Learning via Probabilistic Threshold Estimation . . . . .	35
4	Experiments . . . . .	37
4.1	Experimental Setup . . . . .	37
4.1.1	MIT-BIH Arrhythmia Database . . . . .	37
4.1.2	Patient-specific Domain-adapted Datasets . . . . .	38
4.1.3	Network Training Setup . . . . .	39

4.1.4	Performance Metrics . . . . .	39
4.1.5	Hyperparameters . . . . .	40
4.2	Comparisons Between Representation Errors . . . . .	40
4.3	Personalized and Zero-shot Arrhythmia Detection through Domain Adaptation . . . . .	42
4.4	Ensemble Arrhythmia Detection . . . . .	44
4.5	Comparisons with Competing Methods . . . . .	45
4.6	Energy Efficient Monitoring through Sparse Representation . . . . .	47
5	Conclusion . . . . .	50
	References . . . . .	62

# Preface

First and foremost, I would like to extend my deepest gratitude to my supervisors, Prof. Moncef Gabbouj, Prof. Serkan Kiranyaz, and Dr. Mehmet Yamac, for their invaluable support and guidance throughout my studies. Their expertise and seasoned insights have shaped a clear path for my research and inspired me in this never-ending pursuit of knowledge and discovery.

I am sincerely thankful to the SAMI research group at Tampere University for welcoming me to Finland and sharing their knowledge, experience, and excitement for research with me.

I would like to thank all my friends who shared their support and made this journey not only meaningful but also truly memorable.

Lastly, I am forever grateful to my family for their unwavering support and unconditional love. Their belief in me has been my greatest source of strength and the reason I aspire to improve and dedicate myself fully to everything that I do.

Tampere, December 3, 2024

Mehmet Mert Duman

## List of Figures

2.1	On the left is the structure of a heartbeat during one complete cardiac cycle, consisting of the P wave, PR interval, QRS complex, ST segment, T wave, and QT interval. On the right is the interval between two neighboring R-peaks, known as the R-R interval, which is a measure of the heart rate [45]. Morphological differences in these patterns of the ECG may be indicators of cardiac arrhythmia. . . . .	6
2.2	A single healthy heartbeat and its corresponding beat-trio from patient 100 in the MIT-BIH Arrhythmia Database [48]. . . . .	7
2.3	An example for each type of arrhythmia in the MIT-BIH Arrhythmia Database [48]. . . . .	8
2.4	Examples of ECG segments from four different patients from the MIT-BIH Arrhythmia Database [48]. Each plot shows a segment belonging to a different patient. Heartbeat morphology remains similar for a certain patient (with small variations due to measurement noise and patient movement) but varies significantly between patients. . . . .	9
3.1	The first five atoms of the dictionary for patient (a) in Figure 2.4. The dictionary is learned with Algorithm 1, where $L = 128$ , $K = 20$ , and $\lambda = 0.01$ . . . . .	19
3.2	Example of healthy, S-type, and V-type beats and their corresponding reconstructions. The bottom right plot shows the reconstruction error vectors for all heartbeat types. Both abnormal beats have noticeably large SAE compared to the healthy beat. The beats belong to patient 100 from the MIT-BIH database [48]. . . . .	21
3.3	Overcomplete (a) vs. undercomplete (b) dictionaries for the sparse representation of a signal, $\mathbf{D}\mathbf{x} = \mathbf{s}$ . . . . .	22
3.4	[51]: The four fundamental subspaces and the action of $\mathbf{D}_p$ with rank $r$ . $\mathbf{D}_p$ takes the row space to the column space (blue arrow) and the nullspace to zero (gray arrow). The signal $\mathbf{s}_p^{(i)}$ may not lie in the column space, in which case the representation error emerges and resides in the left nullspace (dotted red line). The transformation, $\mathbf{D}_p\mathbf{x}_p^{(i)}$ (green arrow), can only represent $\hat{\mathbf{s}}_p^{(i)}$ , which is the component of $\mathbf{s}_p^{(i)}$ in the column space (dotted purple line). If $\mathbf{D}_p$ is undercomplete and full rank ( $r = K$ ), there is a unique solution $\mathbf{x}_p^{(j)}$ for any $\hat{\mathbf{s}}_p^{(j)}$ in the column space, as the nullspace is trivial and only contains the zero vector; in other words $\mathbf{x}_p^{(i)} = \mathbf{x}_{p,r}^{(i)}$ and $\mathbf{x}_{p,n}^{(i)} = \mathbf{0}$ . . . . .	23

3.5	Example coefficient vectors $\mathbf{x}_{\ell^1}^{(i)}$ and $\mathbf{x}_{\ell^2}^{(i)}$ for a healthy ECG beat using a dictionary with $K = 20$ atoms. $\mathbf{x}_{\ell^1}^{(i)}$ is more sparse compared to $\mathbf{x}_{\ell^2}^{(i)}$ . . . . .	25
3.6	Symmetrical single hidden-layer autoencoder network architecture for dictionary learning. After training, the learned model weights $\boldsymbol{\theta}_{\mathcal{D}_p}$ correspond to the dictionary. . . . .	27
3.7	Example autoencoder weight matrices, $\boldsymbol{\theta}_{\mathcal{D}_p}\boldsymbol{\theta}_{\mathcal{D}_p}^T$ , acting on $\mathbf{H}_p$ for the two autoencoders. On the left, $\boldsymbol{\theta}_{\mathcal{D}_p}\boldsymbol{\theta}_{\mathcal{D}_p}^T$ is without any activation functions, and on the right, $\boldsymbol{\theta}_{\mathcal{D}_p}\boldsymbol{\theta}_{\mathcal{D}_p}^T$ has a ReLU activation in-between. The bright spots on the diagonal on both matrices correspond to the QRS complex, typically the largest and most prominent wave in the ECG beat. The bottom plot shows the sparsifying effect of ReLU on the coefficients, $\mathbf{X}_p = \boldsymbol{\theta}_{\mathcal{D}_p}\mathbf{H}_p$ , where the coefficients decay faster when ReLU is present (i.e., sparser, in red), and much slower when it is not (in blue). . . . .	28
3.8	Histogram of NPE energies, $\ \tilde{\mathbf{e}}_{\mathcal{N}}\ _2^2$ , for the healthy beats of source, $\mathbf{S}_s$ , and target, $\mathbf{S}_t$ , on the dictionary of target, $\mathbf{D}_t$ . Despite $\mathbf{D}_t$ being learned on healthy ECG beats, it fails to represent the healthy beats of $\mathbf{s}$ due to morphological differences between the two patients' ECGs. . . . .	30
3.9	For a well-learned dictionary for $\mathbf{t}$ , the column space, $C(\mathbf{D}_t)$ , represents a compact subspace for the healthy ECG beats of $\mathbf{t}$ . The projections of $\mathbf{t}$ 's beats onto $C(\mathbf{D}_t)$ , namely $\hat{\mathbf{S}}_t$ , contain most of the components of $\mathbf{S}_t$ , whereas the projections onto $N(\mathbf{D}_t^T)$ —the residuals or representation errors—contain very few components, and have low signal energy, i.e. $\ \mathbf{S}_t - \mathbf{D}_t\mathbf{X}_t\ _2^2 \approx \mathbf{0}$ . The many colors in $C(\mathbf{D}_t)$ and $N(\mathbf{D}_t^T)$ represent different types of signals that also fall onto these subspaces by coincidence. The key point of the figure is that most components of the signals in $\mathbf{S}_t$ fall onto $C(\mathbf{D}_t)$ rather than $N(\mathbf{D}_t^T)$ . . . . .	32
3.10	A visual view of the effect of the transformation $\mathbf{Q}_{s \rightarrow t}$ on $\mathbf{S}_s$ . As shown in Figure 3.8, most of $\mathbf{S}_s$ cannot be represented by $\mathbf{D}_t$ , thus, most of $\mathbf{S}_s$ lives in $N(\mathbf{D}_t^T)$ . After the transformation, the new beats, $\mathbf{S}_{s \rightarrow t}$ , can be represented by $\mathbf{D}_t$ , the dictionary for the healthy beats of $\mathbf{t}$ . Therefore, the transformation creates novel beats that are adapted from $\mathbf{S}_s$ to be morphologically similar to the beats of $\mathbf{t}$ . . . . .	33

3.11	Histogram of NPE energies, $\ \tilde{\mathbf{e}}_N\ _2^2$ , after domain adaptation, for the domain-adapted healthy beats of source, $\mathbf{S}_{s \rightarrow t}$ , and healthy beats of target, $\mathbf{S}_t$ , on the dictionary of target, $\mathbf{D}_t$ . The error energies for the original beats, $\mathbf{S}_s$ , are provided in pale red for comparison. After domain adaptation, the representations of the healthy beats of $\mathbf{s}$ on $\mathbf{D}_t$ are much improved. . . . .	35
3.12	The two-channel convolutional neural network used in all experiments. $K$ is the filter size, and $N$ is the number of neurons in the layer. . . . .	35
4.1	ROC curves of each error type, namely SAE, LAE, NPE, ARE, and ARE (ReLU). The plots indicate that autoencoder representation error with a ReLU activation function separates the healthy and anomalous beats the best. The rest of the error types perform almost equally, with NPE being the fastest to compute among all. . . .	41
4.2	F1-Score vs. threshold of each error type, namely SAE, LAE, NPE, ARE, and ARE (ReLU). The plots show that autoencoder representation error with a ReLU activation function performs the best at $\approx 0.87$ peak F1-Score. The rest of the error types perform equally, with NPE being the least computationally demanding. . . . .	42
4.3	A set of N-type healthy beats from patient 100 vs. R-type healthy beats from patient 232. Despite being healthy, R-type beats have distinct morphological differences from N-type beats. . . . .	44
4.4	The F1-Score of the ensemble classifier for any choice of $\mathcal{C}$ compared against the state-of-the-art arrhythmia synthesis method in [44] and the standalone CNN with domain adaptation. Any choice of $\mathcal{C}$ in a broad range improves the performance over the CNN. . . . .	46
4.5	Energy efficiency over F1-Scores, where the x-axis shows the percentage of test samples that are classified solely based on NPE, hence being more computationally efficient. . . . .	48
4.6	Energy efficiency over F1-Scores, where the x-axis shows the percentage of test samples that are classified solely based on NPE, hence being more computationally efficient. . . . .	48

# List of Tables

3.1	Number of FLOPs needed for calculating different representation errors and the processing time of a single ECG beat on an i7-10870H CPU and an RTX 3080 Laptop GPU. The measurements are averaged over 10 runs and 1000 beats. . . . .	26
4.1	The format of the obtained confusion matrices during evaluation. . .	39
4.2	Cumulative confusion matrices for the baseline (left) and domain-adapted (right) datasets over 10 independent training runs. A = Arrhythmia, H = Healthy. . . . .	43
4.3	Accumulated confusion matrices belonging to patient 232 for the baseline (left) and domain-adapted (right) datasets over 10 independent training runs. A = Arrhythmia, H = Healthy. . . . .	44
4.4	Comparison of the proposed methods, namely RE-C with NPE, Baseline, Domain Adaptation, Ensemble Classification, and Energy-efficient Classification, with the competing methods. The detection performances of prior art, including global and one-shot classifiers, are presented. The results show that our personalized zero-shot ensemble model surpasses all the other methods in F1-Score and is on par with [104], even though [104] is a global one-shot GAN-based classifier with a signal length of 300 (as opposed to 128). The ensemble classifier’s confidence threshold, $\mathcal{C}$ , is chosen using the validation set. The average ensemble classifier is the average performance over all possible confidence thresholds. $\diamond$ Personalized classifiers. * Zero-shot classifiers. . . . .	47

## Abbreviations

AAMI	Association for the Advancement of Medical Instrumentation
ABS	Abnormal Beat Synthesis
ADMM	Alternating Direction Method of Multipliers
AE	Autoencoder
ARE	Autoencoder Reconstruction Error
AUC	Area Under the Curve
BPDN	Basis Pursuit Denoising
CNN	Convolutional Neural Network
CR	Collaborative Representation
CRC	Collaborative Representation-based Classification
DA	Domain Adaptation
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
FLOPS	Floating Point Operations
GAN	Generative Adversarial Network
HRV	Heart Rate Variability
LAE	Least-squares Approximation Error
LSTM	Long Short-term Memory Network
LTI	Linear Time-invariant
MLE	Maximum Likelihood Estimation
MOD	Method of Optimal Directions
MTM	Morphology Transformation Matrix
NN	Neural Network
NPE	Nullspace Projection Error
OMP	Orthogonal Matching Pursuit
PDF	Probability Density Function
RE-C	Representation Error-based Classifier
ROC	Receiver Operating Characteristic
SAE	Sparse Approximation Error
SR	Sparse Representation
SRC	Sparse Representation-based Classification
SVD	Singular Value Decomposition
SVM	Support Vector Machine

# 1 Introduction

An electrocardiogram (ECG) is a painless and non-intrusive recording of the electrical activity in the heart. Since its invention by Willem Einthoven a century ago, ECG has had a substantial impact on clinical practice and has become a fundamental tool in cardiovascular diagnostic procedures [1, 2]. ECG continues to be used routinely to monitor cardiac arrhythmia and various cardiac abnormalities [1, 3], and it has made its way to wearable devices such as the Holter device and smartwatches [4, 5]. For decades, ECGs have been among the most accurate means of detecting heart disturbances and cardiac arrhythmia [1].

Cardiac arrhythmia is an abnormal heart rhythm and can be caused by disruptions in the orderly electrical excitation of the heart [6]. Cardiac arrhythmia often leads to stroke, heart failure, and even sudden cardiac death [7, 8]. In 2018 and 2019, cardiovascular diseases were the leading cause of death globally [9, 10]. According to the 2024 report by the American Heart Association, cardiovascular diseases still result in immense health and economic burdens and cause millions of deaths worldwide [11]. It is thus clear why monitoring and classifying digital electrocardiograms have piqued the interest of many researchers, from signal processing and machine learning communities to medical professionals and healthcare organizations, as advancements in this field can significantly improve the accuracy and efficiency of cardiovascular disease monitoring and diagnosis.

Remarkable progress has been made in ECG classification over the years. Nonetheless, there are many research areas open to improvement and areas that are yet to be explored. This thesis focuses on real-time and efficient monitoring of cardiac arrhythmia suitable for low-performance computing systems such as wearable devices, and addresses two important challenges that arise when healthy individuals want to benefit from continuous monitoring: (i) lack of data for effective learning using machine learning methods, and (ii) discrepancies in ECG morphology when existing data is reused for ECG classification across different patients. The first challenge arises because healthy individuals typically do not experience arrhythmia, leading to a scarcity of relevant ECG data to train machine learning-based classifiers. The second challenge is due to differences in physiological characteristics among patients, which leads to morphological differences in their ECG, making it difficult to generalize models across different individuals [12, 13]. In fact, it is possible to perform robust person identification from ECG due to these morphological differences, and researchers have extensively studied and developed various methods on this topic [14, 15, 16, 17].

A vast amount of literature exists on ECG analysis in the machine learning

community. Main research directions in the literature can be categorized into person identification, heartbeat detection (i.e., beat segmentation), beat classification, and beat generation. The first stage of analysing an ECG typically starts with finding the individual heartbeats. Beat detection is critical as any error made in this step will naturally propagate to subsequent steps and hinder the quality of analysis [18]. ECGs have several key segments corresponding to different phases of the heart’s electrical cycle [1]. One of those components, the QRS complex, is generally the largest wave in the ECG for a healthy beat, and the peak of this wave is called the R-peak. The primary objective in beat detection is to find the R-peaks in a given ECG band. Pan and Tompkins developed one of the most widely used R-peak detection algorithms and set the baseline for heartbeat detection [19]. Since then, numerous advancements have been proposed for accurate R-peak detection [20, 21, 22], including segmentation of challenging low-quality Holter ECGs [23].

The next step after heartbeat detection is to process and classify the beats as healthy or as one of the several arrhythmia types. A wide range of methods have been developed for beat classification, including those based on hand-crafted or manually extracted ECG features [24, 25, 26, 27, 28], statistical analysis [29, 30], frequency-domain analysis [31, 32, 33], and several machine learning algorithms like k-nearest neighbors and support vector machines [34, 35]. Early approaches have focused on training a single classifier to detect arrhythmias across all patients (i.e., global classifiers), which soon realized that ECG classification is a highly personalized problem. Subsequent works have shifted towards personalized classifiers, as seen in works such as [36, 37, 38]. However, the proposed classifiers relied on large amounts of labeled healthy and anomalous beats from the patient during training, limiting their applicability when dealing with healthy individuals who lack sufficient abnormal data.

Researchers have explored methods for generating synthetic ECG beats to overcome the lack of sufficient arrhythmia in healthy individuals. Beat generation is closely tied to beat classification, as beat generation serves as a way to improve classification performance. Among the proposed methods, various Generative Adversarial Network (GAN) architectures have been developed to synthesize both healthy and arrhythmic beats [39, 40, 41, 42, 43]. These methods primarily address the data imbalance problem inherent in most ECG datasets rather than enabling zero-shot arrhythmia detection—where the classifier has no prior exposure to the individual’s arrhythmia during training. Nonetheless, generative techniques enable more robust personalized classifiers to be trained by enriching training datasets with artificially generated beats. In a landmark study [44], Kiranyaz *et al.* proposed a personalized beat generation method called *abnormal beat synthesis* (ABS). In their work, the authors model the degradation from healthy beats to abnormal beats as a linear

time-invariant (LTI) system and obtain a filter bank by modeling several common causes of heart disease. These filters are then used to generate anomalous beats directly from healthy beats.

Drawing inspiration from innovations such as ABS, this thesis contributes to the field by introducing a novel sparse representation-based (SR) domain adaptation (DA) method for inter-patient beat transfer. Our approach eliminates the need to generate synthetic beats altogether; instead, it addresses the arrhythmia scarcity problem by transferring real anomalous beats from other individuals into the target patient’s domain, enriching their healthy beat database with original domain-adapted beats from other individuals and thus enhancing personalized arrhythmia detection.

Then, this thesis proposes a practical and robust arrhythmia monitoring system through an ensemble classification model. The first part of the ensemble model is a representation error-based lightweight classifier (RE-C) built upon the proposed nullspace projection error (NPE) that confidently classifies a large portion of the healthy beats immediately from their NPE. The rest of the beats are forwarded to a more sophisticated convolutional neural network (CNN), which is trained on a small set of healthy beats belonging to the individual and additional domain-adapted beats from others. The suspicious beats are passed through the CNN, which predicts the probability of each beat being healthy. Based on the confidence level of the CNN, the final classification is determined either by the RE-C or the CNN.

## 1.1 Motivation

Electrocardiograms provide a non-intrusive and accurate way to monitor cardiac arrhythmia and remain indispensable in modern healthcare. Given that cardiovascular diseases are still the leading cause of death worldwide, any advancements in ECG analysis can have a profound impact on the early detection of cardiac arrhythmia, and prevent life-threatening conditions such as stroke, heart failure, and sudden cardiac death. Improvements in arrhythmia detection, particularly through personalized and zero-shot monitoring systems, have the potential to be employed in real-life conditions and deliver substantial health benefits. As such, this field of study continues to be highly important in healthcare, and every improvement offers a chance to impact millions of lives.

## 1.2 Contributions

The contributions of this thesis can be summarized as follows:

- **Complete Pipeline for Personalized and Zero-shot ECG Monitoring:**  
This thesis proposes a comprehensive pipeline for personalized and zero-shot

ECG arrhythmia detection, enabling continuous monitoring for healthy individuals with minimal healthy and no anomalous ECG data.

- **Efficient Representation Error-based Detection:** Leveraging linear algebra, we propose an efficient representation error-based arrhythmia detection method using nullspace projection error (NPE) and least-squares approximation error (LAE), reducing computational costs significantly while maintaining performance.
- **Compact Dictionary Learning via Autoencoders:** Through single hidden-layer autoencoders, we learn compact dictionaries, reducing computational complexity and improving representation error-based arrhythmia detection.
- **Sparse Representation-based Domain Adaptation:** We propose an SR-based domain adaptation method through morphology transformation matrices (MTMs), enabling inter-patient beat transfer and substantial performance improvements without synthetic data generation.
- **Ensemble Arrhythmia Detection:** We design an ensemble classifier by combining RE-C and CNN models, improving detection performance and achieving state-of-the-art accuracy and F1-scores.
- **Energy-efficient Arrhythmia Monitoring:** We present an energy-efficient monitoring system by leveraging NPE for efficient arrhythmia detection, suitable for mobile devices and capable of reducing energy consumption without sacrificing performance.

### 1.3 Thesis Structure

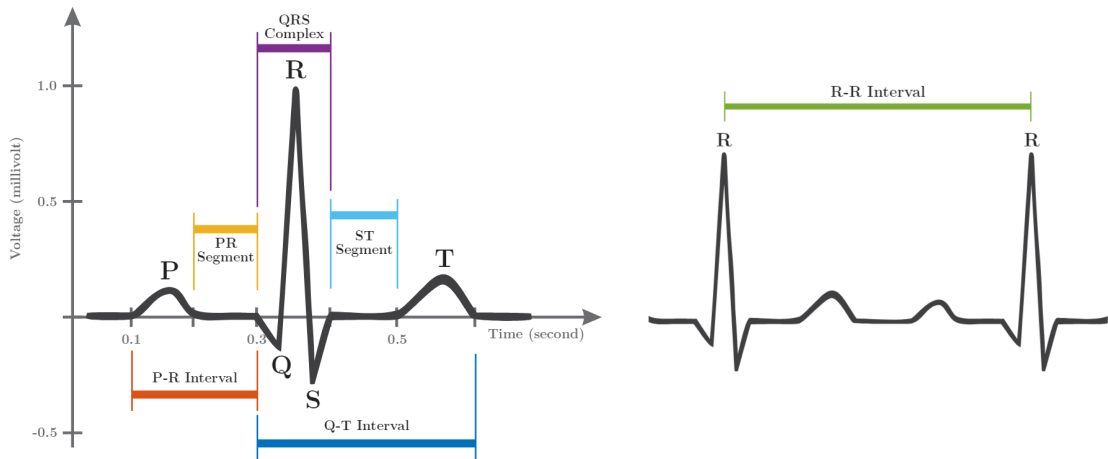
The rest of the thesis is organized as follows: Chapter 2 presents the necessary background, including a brief introduction to sparse representation theory, a review of relevant literature on sparse representation-based classification, and methods for cardiac arrhythmia detection and synthesis. Chapter 3 describes the methods developed and employed in this research. This includes computationally efficient representation error-based classification, SR-based domain adaptation, and the development of a robust ECG monitoring system. Chapter 4 details the experiments conducted to evaluate the proposed methods, including comparisons with existing literature. Finally, Chapter 5 concludes the thesis, summarizing the findings and suggesting directions for future research.

## 2 Background

### 2.1 Electrocardiogram

#### 2.1.1 Salient Characteristics of ECG

An ECG captures the heart's electrical activity over time as a series of voltage changes. These voltage changes form distinctive structures that characterize the ECG. Figure 2.1 shows a complete cardiac cycle of a typical healthy heartbeat. The ECG consists of several patterns: the P wave, PR interval, QRS complex, ST segment, T wave, and QT interval, each corresponding to specific phases of cardiac activity. Severe morphological differences in these patterns or their absence usually indicate cardiac abnormalities.

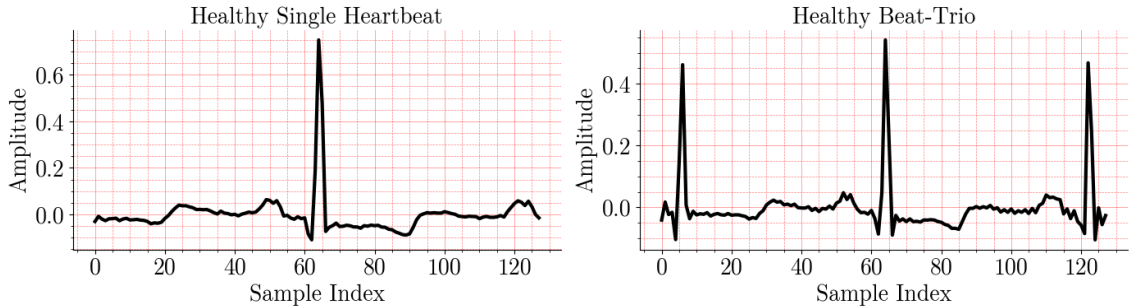


**Figure 2.1** On the left is the structure of a heartbeat during one complete cardiac cycle, consisting of the P wave, PR interval, QRS complex, ST segment, T wave, and QT interval. On the right is the interval between two neighboring R-peaks, known as the R-R interval, which is a measure of the heart rate [45]. Morphological differences in these patterns of the ECG may be indicators of cardiac arrhythmia.

The most prominent of the features of a healthy ECG is typically the QRS complex, which begins with a negative deflection, followed by a rapid spike, and ends with another negative deflection. The peak of the QRS complex, namely the R-peak, is usually the largest wave in the ECG [1]. Due to this property, most beat detection algorithms, such as Pan and Tompkins [19], work by detecting the R-peaks in a band of ECG.

Another important feature in ECG analysis is the R-R interval, which measures the time between two consecutive R-peaks. The R-R interval serves as an indicator of heart rate, and its variability, known as heart rate variability (HRV), is also a

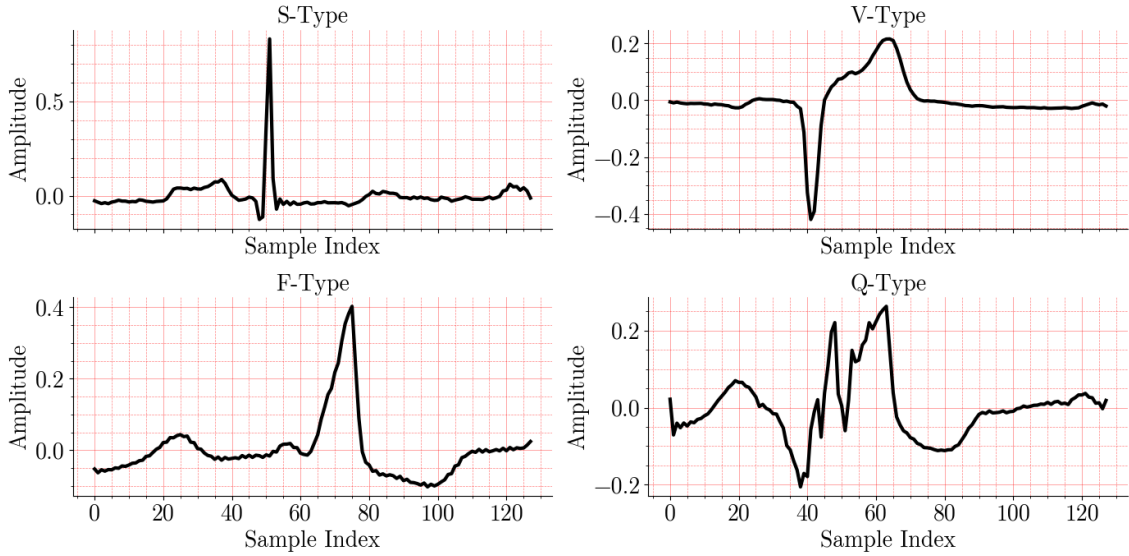
key diagnostic feature often used to assess overall cardiovascular health [46]. Steady R-R intervals are a sign of a regular heartbeat, while irregular intervals may suggest arrhythmias or other abnormalities in heart rhythm [47]. Accurate analysis of the R-R interval is thus essential in identifying both normal and anomalous heartbeats in clinical practice. To this end, in Chapter 3, we use single heartbeats and additionally include their neighboring R-peaks (together called a beat-trio) when building our arrhythmia detection system. An example of a single beat and its corresponding beat-trio is given in Figure 2.2.



**Figure 2.2** A single healthy heartbeat and its corresponding beat-trio from patient 100 in the MIT-BIH Arrhythmia Database [48].

### 2.1.2 Categories of ECG Arrhythmia

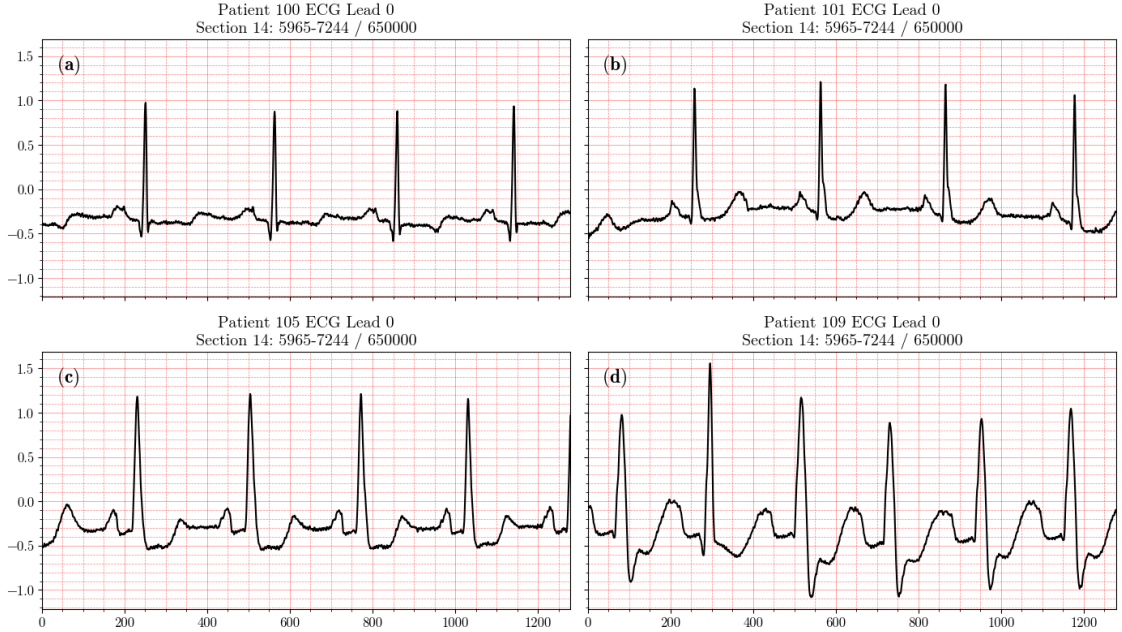
The Association for the Advancement of Medical Instrumentation (AAMI) recommends combining anomalous heartbeats under the following four heartbeat types: ventricular ectopic beats (V or VEB), supraventricular ectopic beats (S or SVEB), fusion beats (F), and uncategorizable beats (Q). S-type beats are premature heartbeats, and their R-peaks occur earlier than the next expected normal R-peak. V-type beats are identified by their broad, abnormal QRS complexes. VEBs do not have preceding P waves, and the QRS complex is often followed by a full compensatory pause. F-type beats appear as intermediate complexes that are part normal (narrow) and part ectopic (wide). They create a hybrid appearance between normal and V-type beats. Finally, Q-type beats have unclear morphology and do not follow typical patterns [1]. Figure 2.3 presents an example for each type of arrhythmia in the MIT-BIH database.



**Figure 2.3** An example for each type of arrhythmia in the MIT-BIH Arrhythmia Database [48].

### 2.1.3 Personalized and Zero-shot Arrhythmia Detection

ECG patterns may exhibit significant variations across individuals due to physiological differences, making ECG signals highly personalized [12, 13]. This variability poses a considerable challenge for classification models, as they often struggle to generalize across different patients. Figure 2.4 shows small ECG segments from 4 patients in the MIT-BIH Arrhythmia Database [48]. Although the morphology of heartbeats within each patient’s ECG remains relatively consistent, there are pronounced differences across different patients. These morphological variations between patients complicate, or even prevent, the reuse of ECG data for one patient to train an arrhythmia detection system for another patient. This necessitates the development of zero-shot learning approaches, which aim to robustly detect arrhythmias in previously unseen data without requiring patient-specific anomalous training examples.



**Figure 2.4** Examples of ECG segments from four different patients from the MIT-BIH Arrhythmia Database [48]. Each plot shows a segment belonging to a different patient. Heartbeat morphology remains similar for a certain patient (with small variations due to measurement noise and patient movement) but varies significantly between patients.

## 2.2 Sparse Representation-based Classification

### 2.2.1 Notations

Throughout this thesis, the  $p$ -norm (i.e.  $\ell^p$  norm) of a vector  $\mathbf{x} \in \mathbb{R}^K$  for  $p \geq 1$  is defined as

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^K |x_i|^p \right)^{1/p}, \quad (2.1)$$

while the special case, the  $\ell^0$  norm, is defined as  $\|\mathbf{x}\|_0 = \#\{i \mid x_i \neq 0\}$ , which counts the number of non-zero elements of  $\mathbf{x}$ . Similarly, the  $p$ -norm of a matrix  $\mathbf{X} \in \mathbb{R}^{K \times B}$  for  $p \geq 1$  is defined as

$$\|\mathbf{X}\|_p = \left( \sum_{i=1}^K \sum_{j=1}^B |X_{ij}|^p \right)^{1/p}. \quad (2.2)$$

Finally,  $\mathbf{X}_{\cdot i}$  denotes the  $i$ -th column, and  $\mathbf{X}_i$  denotes the  $i$ -th row of a matrix  $\mathbf{X}$ .

### 2.2.2 Sparse Representation

Sparse representation (SR) is a fundamental concept in signal processing and machine learning. SR is rooted in the idea that many signals—such as images and audio signals—can be represented by a linear combination of a small number of

atoms from a dictionary. Consider a linear system of equations

$$\mathbf{s} = \mathbf{D}\mathbf{x}, \quad (2.3)$$

where  $\mathbf{s} \in \mathbb{R}^L$  is a measured signal,  $\mathbf{D} \in \mathbb{R}^{L \times K}$  is a dictionary of atoms, and  $\mathbf{x} \in \mathbb{R}^K$  is the coefficient vector. For simplicity, we can assume that  $\mathbf{D}$  is a full rank matrix and eliminate the case of having no solutions. If the system is determined ( $K = L$ ), each signal is uniquely represented in  $\mathbf{D}$  and we can find the coefficient vector as  $\mathbf{x} = \mathbf{D}^{-1}\mathbf{s}$ . In the case of an underdetermined system ( $K > L$ ), the set of all solutions forms a  $K - L$  dimensional subspace in  $\mathbb{R}^K$ , and thus there are infinitely many  $\mathbf{x}$  that satisfy 2.3 [49]. We can introduce a cost function,  $J$ , to evaluate each solution and narrow down the solutions, possibly finding a unique one. Thus, we define the following optimization problem:

$$\min_{\mathbf{x}} J(\mathbf{x}) \quad \text{subject to } \mathbf{s} = \mathbf{D}\mathbf{x}. \quad (2.4)$$

A standard choice for  $J$  is the  $p$ -norm (or  $\ell^p$  norm) as the  $\ell^p$  norm is a convex function of  $\mathbf{x}$  for  $p \geq 1$  [50]. In particular, for  $p > 1$ , the  $\ell^p$  norm is strictly convex, which ensures that a unique solution exists. The  $p$ -norm minimization problem can be formulated as

$$\min_{\mathbf{x}} \|\mathbf{x}\|_p \quad \text{subject to } \mathbf{s} = \mathbf{D}\mathbf{x}. \quad (2.5)$$

It might be tempting to minimize the  $\ell^2$  norm as it yields a simple closed-form solution  $\mathbf{x}^+ = \mathbf{D}^+\mathbf{s}$  through the pseudoinverse of  $\mathbf{D}$ . If  $\mathbf{D}$  is full rank, then  $\mathbf{D}^+ = \mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1}$ . Otherwise, we can find  $\mathbf{D}^+ = \mathbf{V}\Sigma^+\mathbf{U}^T$  through the singular value decomposition (SVD) of  $\mathbf{D}$  [51]. However,  $\mathbf{x}^+$  is typically not sparse and often has many small components [49].

To promote sparsity, one could minimize the  $\ell^0$  norm instead, aiming to represent  $\mathbf{s}$  with only a small number of non-zero components:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to } \mathbf{s} = \mathbf{D}\mathbf{x}. \quad (2.6)$$

Problem 2.6 is called the *sparse representation* of the signal  $\mathbf{s}$ . However,  $\ell^0$  norm lacks the desirable properties of  $\ell^2$  norm:  $\ell^0$  norm is neither convex nor an actual norm (since  $\|c\mathbf{x}\|_0 = \|\mathbf{x}\|_0 \neq |c|\|\mathbf{x}\|_0$  for non-zero scalar  $c$ ). No polynomial-time algorithm exists to solve Problem 2.6; it is NP-Hard. In the literature, there are several studies on greedy algorithms that provide approximate solutions to 2.6, like *Matching Pursuit* [52] and *Orthogonal Matching Pursuit* (OMP) [53]. The convergence of these algorithms can be guaranteed for sufficiently sparse solutions [49, 54]. Other methods for tackling 2.6 include relaxing the  $\ell^0$  norm with its closest

convex counterpart, the  $\ell^1$  norm, called *Basis Pursuit* [55]. Moreover, in practice, the equality condition in 2.6 is too strict and is often relaxed to allow for small deviations in the representation:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{s} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon. \quad (2.7)$$

This problem is referred to as *sparse coding* in many dictionary learning algorithms [49, 56]. Similar to *Basis Pursuit*, there are algorithms designed to solve the convex-relaxed variant of 2.7, known as *Basis Pursuit Denoising* (BPDN) [55] in the literature. A closely related optimization problem to 2.7 is the Lasso [57], which also seeks a sparse solution but introduces a regularization parameter  $\lambda$  to balance the trade-off between the approximation error and sparsity:

$$\min_{\mathbf{x}} \|\mathbf{s} - \mathbf{D}\mathbf{x}\|_2 + \lambda \|\mathbf{x}\|_1. \quad (2.8)$$

In sparse representation, dictionaries are often overcomplete ( $K > L$ ) to enhance their ability to represent a broader range of signals with fewer atoms. In contrast, when the linear system is overdetermined ( $K < L$ ), the column vectors of the dictionary cannot span the entire signal space  $\mathbb{R}^L$ , and the signal  $\mathbf{s}$  may lie outside the *column space of  $\mathbf{D}$* , denoted as  $C(\mathbf{D})$ . Nonetheless, 2.7 is still valid for undercomplete dictionaries as it permits close approximations of the signal. In Chapter 3, we go into detail about such systems and undercomplete dictionaries.

### 2.2.3 Dictionary Learning

A fundamental part of sparse representation is determining a proper dictionary  $\mathbf{D}$ . In early literature, the popularity of signal transforms played a significant role in dictionary design. These transforms, such as the Fourier Transform, Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT), naturally led to the creation of the often called *analysis dictionaries* using their respective bases [58]. Analysis dictionaries are simple and often fast; however, they have their shortcomings: Fourier basis and DCT generate dense coefficient vectors in the presence of discontinuities or localized features in the signal, and DWT depends heavily on the choice of the wavelet function. In recent years, the direction of research has shifted towards dictionary learning to overcome these shortcomings [59, 60, 61, 62]. Dictionary learning adapts the dictionary to the specific patterns of the signal, leading to sparser and more accurate representations.

Dictionary learning often starts with a set of signals  $\mathbf{S} \in \mathbb{R}^{L \times B}$  of some arbitrary size  $B$ . The goal is to find a dictionary  $\mathbf{D}$ , along with a matrix of coefficients  $\mathbf{X} \in \mathbb{R}^{K \times B}$ , that satisfy the following optimization problem:

$$\min_{\mathbf{D}, \mathbf{X}} \|\mathbf{X}\|_0 \quad \text{subject to} \quad \|\mathbf{S} - \mathbf{D}\mathbf{X}\|_2 \leq \epsilon. \quad (2.9)$$

A strategy to address this optimization problem is to iteratively optimize the dictionary  $\mathbf{D}^{\{t\}}$  and the coefficient matrix  $\mathbf{X}^{\{t\}}$  at each iteration  $t$  in an alternating fashion. At each iteration,  $\mathbf{D}^{\{t-1\}}$  from the previous iteration is used in solving 2.7 to find the sparse codes  $\mathbf{x}_i$  for each sample  $\mathbf{s}_i$  with a basis pursuit algorithm. These sparse codes are then stacked to form the updated coefficient matrix  $\mathbf{X}^{\{t\}}$ . Then, we fix  $\mathbf{X}^{\{t\}}$  and solve for  $\mathbf{D}^{\{t\}}$ . The optimal  $\mathbf{D}^{\{t\}}$  at each iteration is given by  $\mathbf{D}^{\{t\}} = \mathbf{S}\mathbf{X}^{\{t\}+}$ . This algorithm is called the *Method of Optimal Directions* (MOD) [63]. Another such well-known algorithm is K-SVD [64], which sequentially learns the atoms in the dictionary one by one instead of learning the entire dictionary at once. An array of algorithms have been designed for dictionary learning using different strategies, such as online dictionary learning [56], training feed-forward encoder architectures for sparse code prediction [65], algorithms based on *FOCal Underdetermined System Solver* [66, 67] and its variants [68, 69], and more [58].

## Deep Dictionary Learning

Following the rise in popularity of machine learning and deep learning methods, many researchers have demonstrated that dictionary learning is possible through deep neural networks (NN). There is a vast body of literature dedicated to training various neural network architectures for the dictionary learning task, including Restricted Boltzmann Machines [70, 71, 72], Variational and Sparse Autoencoders (AE) [73, 74], and Convolutional Neural Networks [75, 76]. This thesis is interested in training autoencoders for dictionary learning, which boils down to the following generalized approach:

$$\hat{\mathbf{E}}, \hat{\mathcal{D}} = \arg \min_{\mathbf{E}, \mathcal{D}} \mathcal{L}(\mathbf{S}, \mathcal{D}(\phi(\mathbf{E}(\mathbf{S})))) + J(\mathbf{E}, \mathbf{S}) + R_1(\boldsymbol{\theta}_{\mathcal{D}}) + R_2(\boldsymbol{\theta}_{\mathbf{E}}), \quad (2.10)$$

where  $\mathcal{D}$  is the dictionary (otherwise called the *decoder*),  $\mathbf{E}$  is the encoder,  $\mathcal{L}$  is the penalty on reconstruction error,  $J$  is the penalty on encoder activations, and  $R_1, R_2$  are regularization terms for the model weights,  $\boldsymbol{\theta}_{\mathcal{D}}, \boldsymbol{\theta}_{\mathbf{E}}$ .  $\phi$  is an activation function applied after the encoder, which is typically the ReLU activation, but it can also be omitted.

In the most classical case, we have a single hidden-layer sparse autoencoder, where  $\boldsymbol{\theta}_{\mathbf{E}} \in \mathbb{R}^{K \times L}$  and  $\boldsymbol{\theta}_{\mathcal{D}} \in \mathbb{R}^{L \times K}$ . It is common to set  $\mathcal{L}$  as the  $\ell^2$  norm of the original and reconstructed signals, and  $J$  as the  $\ell^1$  norm of the encoder activations to promote sparsity [74, 77], similar to Section 2.2.3. Weights may be regularized

through weight decay for robustness [78], i.e.,  $R_1 = R_2 = \|\cdot\|_F^2$ . This yields the following optimization problem to find the model weights,  $\boldsymbol{\theta}_E$  and  $\boldsymbol{\theta}_D$ , which is typically optimized through backpropagation:

$$\hat{\boldsymbol{\theta}}_E, \hat{\boldsymbol{\theta}}_D = \arg \min_{\boldsymbol{\theta}_E, \boldsymbol{\theta}_D} \|\mathbf{S} - \boldsymbol{\theta}_D \boldsymbol{\theta}_E \mathbf{S}\|_2 + \|\boldsymbol{\theta}_E \mathbf{S}\|_1 + \|\boldsymbol{\theta}_D\|_F^2 + \|\boldsymbol{\theta}_E\|_F^2, \quad (2.11)$$

where  $\boldsymbol{\theta}_E \mathbf{S}$  corresponds to the coefficient matrix  $\mathbf{X}$  of  $\boldsymbol{\theta}_D$  and  $\phi$  is omitted. The depth and complexity of the encoder and decoder models,  $E$  and  $D$ , have drastically grown as the dataset sizes increased and problems turned more difficult. On this account, deep dictionary learning has been used to address many challenging problems such as ECG classification [79], face, object, and handwritten digit recognition [80], machine fault diagnosis [81], image denoising [75], and for support estimation to aid in many classification and regression tasks [76].

#### 2.2.4 Sparse and Collaborative Representation-based Classification

Conventional sparse representation-based classification (SRC) deals with overcomplete dictionaries learned from a set of labeled training signals with multiple distinct classes [82]. Overcomplete dictionaries are desirable, as they allow for more flexibility in representing signals, leading to sparser and more robust representations. In conventional SRC, we have a supervised training scheme where the overcomplete dictionary  $D$  is composed by learning and concatenating multiple sub-dictionaries  $D_{c_i}$  for each class  $c_i$ :

$$D = \begin{bmatrix} D_{c_1} & D_{c_2} & \cdots & D_{c_n} \end{bmatrix}. \quad (2.12)$$

Each sub-dictionary  $D_{c_i}$  can be learned through one of the methods described in Section 2.2.3 with a sub-set of the training signals  $\mathbf{S}_{c_i}$  corresponding to class  $c_i$ . In their seminal paper [82], Wright *et al.* form each sub-dictionary  $D_{c_i}$  by downsampling and arranging the training signals  $\mathbf{S}_{c_i}$  as the columns of the matrix  $D_{c_i}$ . This simple method does indeed form a dictionary that achieves sparse representations; however, learning the dictionary can provide better generalization by capturing the essential features and structures of the data. Moreover, the former method results in large dictionaries that linearly scale with the size of the training set, whereas a learned dictionary can be compact, depending on the number of atoms desired.

In the test stage, given a new signal  $\mathbf{s}_u$  from an unknown class, its sparse coefficients can be obtained through BPDN or a greedy algorithm like OMP. After obtaining the sparse coefficient vector  $\mathbf{x}_u$ , the residual error is computed for each

class:

$$r_{c_i}(\mathbf{s}_u) = \|\mathbf{s}_u - \mathbf{D}_{c_i}\mathbf{x}_{c_i}\|_2, \quad (2.13)$$

where  $\mathbf{x}_{c_i}$  is the part of  $\mathbf{x}_u$  corresponding to class  $c_i$ . Finally, we can classify the test signal into the class that minimizes the residual error:

$$\text{Class}(\mathbf{s}_u) = c_i^* = \arg \min_{c_i} r_{c_i}(\mathbf{s}_u). \quad (2.14)$$

SRC has proven to be a powerful tool in various applications, including face recognition [82], hyperspectral image classification [83], audio classification [84], and brain tumor diagnosis [85], demonstrating its effectiveness in handling diverse classification tasks.

Conventional SRC is a multiclass classification problem, where the coefficient vector  $\mathbf{x}$  is calculated over the entire dictionary  $\mathbf{D}$  that is comprised of several class-specific sub-dictionaries  $\mathbf{D}_{c_i}$ . The method of using training samples (or learned atoms) from all classes to represent the test signal  $\mathbf{s}_u$  is called *collaborative representation* (CR) [86]. The authors of [86] show that CR-based classification (CRC), which uses  $\ell^2$  regularized coefficients instead of  $\ell^1$ , works as well as SRC in the canonical problem of face recognition due to both being collaborative representation methods. Unlike SRC and CRC, the central problem of this thesis, personalized and zero-shot ECG arrhythmia classification, is a one-class classification problem where the dictionary atoms are learned from only a single class. Therefore, this thesis focuses on SR for anomaly (or outlier) detection schemes, which are not collaborative representations.

### 2.2.5 Sparse Representation for ECG Classification

Sparse representation has been successfully used in several ECG classification tasks, such as [87, 88]. However, these methods construct the dictionaries by extracting the features of the ECG (see Section 2.1.1) rather than directly from the raw signal. Therefore, their performance is limited by the effectiveness of manual feature extraction methods. On the other hand, in [89] and [90], the authors propose learning the dictionary directly from the healthy heartbeats of a specific patient.

In [89], Carrera *et al.* utilize SRC for personalized zero-shot ECG arrhythmia classification on the MIT-BIH Arrhythmia Database [48, 91]. The authors gather 500 healthy ECG beats and learn patient-specific dictionaries,  $\mathbf{D}_p \in \mathbb{R}^{L \times K}$ , for each patient  $p \in [1, M]$  using the K-SVD [64] algorithm mentioned in Section 2.2.3. In contrast to conventional SRC, the dictionaries are undercomplete ( $K < L$ ) and represent only a single class: healthy ECG beats for patient  $p$ . To reiterate, the optimization problem 2.7 to find a sparse solution is still valid even if a signal

belonging to patient  $p$  is not in the column space of  $\mathbf{D}_p$  as it solves for the closest approximation of the signal in  $\mathbf{D}_p$  instead of an exact solution.

To detect whether a new signal  $\mathbf{s}_p^{(i)}$  belonging to patient  $p$  is an arrhythmia or not, Carrera *et al.* solve 2.7 by adopting the greedy and iterative OMP algorithm, and obtain the sparse coefficient vector  $\mathbf{x}_p^{(i)}$ . Then, the residual error is computed as

$$r_p(\mathbf{s}_p^{(i)}) = \|\tilde{\mathbf{e}}_{\ell^1}\|_2 = \|\mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{x}_p^{(i)}\|_2. \quad (2.15)$$

The residual error is used to discriminate whether  $\mathbf{s}_p^{(i)}$  belongs to the healthy beat space of  $p$  or not. The authors conclude that large values of  $r_p(\mathbf{s}_p^{(i)})$  indicate a signal with a different morphology than the healthy beats of  $p$ . Thus, Carrera *et al.* empirically find a threshold  $\tau$  and classify any beat whose residual error is greater than  $\tau$  as anomalous [89]. In a subsequent study [92], Carrera *et al.* propose a dictionary and threshold adaptation mechanism based on the changes in the patient’s heart rate for long-term ECG monitoring.

Throughout the thesis, if the coefficient vector  $\mathbf{x}_p^{(i)}$  is sparse, we refer to its corresponding residual error vector as the **sparse approximation error (SAE)**, denoted with  $\tilde{\mathbf{e}}_{\ell^1}$ . SAE-based classification is a special case of SRC, using only a single class and an empirical threshold for classification. On the other hand, SRC is a multiclass problem, where a given signal is classified as the class that results in minimum residual error.

### 2.3 Methods for Cardiac Arrhythmia Detection

There is a vast amount of literature for cardiac arrhythmia detection using signal processing and machine learning methods, including statistical analysis [29, 30], frequency analysis [32, 31, 33], self-organizing maps [93], k-nearest neighbor classifiers [34], and support vector machines (SVM) [35]. In [24], Chazal *et al.* propose a method to extract the salient features of the ECG, such as the R-R interval, T-wave offset, QRS onset and offset, and fit a linear discriminant classifier to detect arrhythmia. They fit a single global classifier, i.e., a classifier for all patients in the MIT-BIH database. In their follow-up work [25], Chazal *et al.* improve the global classifier by fitting patient-specific linear discriminant models for each patient. Several other feature extraction-based methods exist that utilize linear discriminant analysis for ECG classification [26, 27, 28]. Due to relying on handcrafted features, most of these methods are not robust to the variations in ECG patterns.

Several ML paradigms have been proposed for ECG classification, including Convolutional Neural Networks (CNN) [94, 95], Long Short-term Memory Networks (LSTM) [96], and Transfer Learning [97]. In [98], the authors train a 2-D CNN

for arrhythmia classification from images of ECG bands. In [99], once again, various features of the ECG are extracted (e.g., R-R intervals, beat-to-beat correlation, etc.). Then, the features are stacked to create 3-D inputs and fed into a 3-D CNN for classification. The main pitfall of these methods is training a single global classifier, which does not generalize well to the physiological changes between different patients.

A personalized classification system is proposed in [36], which builds a block-based CNN structure for arrhythmia detection. In [37], Kiranyaz *et al.* conduct a pivotal study on using 1-D CNNs for personalized arrhythmia detection. In a later publication [38], the authors improve their results and propose a novel neuron model called *generative neurons*. In [100], Zhai *et al.* represent ECG signals as 2-D matrices by taking the outer product of two adjacent dual-beats (i.e., two pairs of adjacent beats). This so-called dual-beat coupling matrix captures the temporal morphological characteristics of the adjacent beats. Then, a 2-D CNN is trained on the coupling matrices in a supervised fashion for personalized arrhythmia detection. Despite performing robustly and offering unique perspectives on arrhythmia detection, these methods rely on both healthy and anomalous heartbeat data from the patient. Therefore, such methods cannot be used as early detection systems for a healthy individual for whom we have no anomalous ECG data to train a classifier.

## 2.4 Arrhythmia Synthesis for Detection

Arrhythmia synthesis has garnered considerable attention from researchers focusing on ECG classification and arrhythmia detection. This interest stems from the fact that a healthy individual with no history of cardiac arrhythmia does not have anomalous ECG beats. As discussed earlier (e.g., see Section 2.1.3), reusing ECG beats from different patients often fails due to morphological variations between each individual’s heartbeats. Consequently, the ability to generate synthetic arrhythmia directly helps in training dedicated classifiers for such healthy individuals. In [44], Kiranyaz *et al.* contribute a landmark study. In their work, the authors propose a beat generation method named *abnormal beat synthesis* (ABS). The idea is to model the degradation from healthy to abnormal beats as a linear time-invariant (LTI) system. Then, a filter bank is obtained by modeling several common causes of heart disease through the LTI system. The degradations are modeled as follows. First, the authors find the average healthy beat for a patient  $p$  in the system, denoted as  $\bar{\mathbf{s}}_p \in \mathbb{R}^L$ . This beat can be found by averaging all the healthy beats and finding the closest beat to the average. Then, for each anomalous beat  $\mathbf{s}_p^{(A_i)}$  belonging to  $p$ , the authors estimate a filter  $\mathbf{h}_p^{(i)} \in \mathbb{R}^L$  such that:

$$\bar{\mathbf{s}}_p * \mathbf{h}_p^{(i)} = \mathbf{s}_p^{(A_i)}. \quad (2.16)$$

In other words, the filter  $\mathbf{h}_p^{(i)}$  is convolved with the average healthy beat to produce its corresponding arrhythmia. By solving Equation 2.16 through regularized least-squares filtering, the authors show that abnormal heartbeats can be modeled as linearly degraded versions of healthy heartbeats. After obtaining all filters, similar ones are pruned, and the remaining filters form the filter bank for patient  $p$ . When a new individual  $t$  registers to the system, the authors collect a small set of heartbeats from  $t$  and find their average healthy beat,  $\bar{\mathbf{s}}_t$ . Then, any filter  $\mathbf{h}_p^{(i)}$  from any of the patients already registered to the ABS system can be applied to  $\bar{\mathbf{s}}_t$  to generate abnormal beats. Using these filter banks, Kiranyaz *et al.* propose an innovative beat generation method that can supply healthy individuals with synthetic abnormal beats, allowing compact and lightweight classifiers to be trained, such as a 3-layer 1D CNN.

### 2.4.1 GAN-based Synthesizers

In the related literature for ECG classification and arrhythmia detection, GANs are widely used for beat generation and classification. To provide a fair comparison with the work presented in this thesis, the related studies are categorized into four groups: (i) data generator GANs, (ii) vanilla GANs, (iii) personalized GANs, and (iv) personalized and zero-shot GANs.

**Data generator GANs** focus on generating beats that are similar to the original beats from the patient, but they do not perform arrhythmia detection. In [101], the authors train several GAN architectures, combining LSTM and bidirectional LSTM generators with CNN discriminators to produce synthetic beats. They evaluate the quality of their generations using metrics such as maximum mean discrepancy (MMD) and dynamic time warping (DTW). In another study, Wulan *et al.* evaluate the quality of the generated beats with two novel measures, namely *GAN-train* and *GAN-test* [102]. *GAN-train* is calculated by training an SVM on the generated beats and testing on the real beats, while *GAN-test* is calculated by training the SVM on the real beats and testing on the generated beats. The performance of the SVM on both metrics reflects the similarity between the generated and real beat distributions. In [103], Zhu *et al.* train bidirectional LSTM generators and CNN discriminators similar to [101], but they evaluate their generations using Frechet distance (FD), mean square distance (PRD), and root mean square error (RMSE).

**Vanilla GANs** serve as both beat generators and classifiers for the entire database of patients, where the model is trained *including* each individual’s abnormal beats. A global GAN model generates abnormal beats akin to all anomalies in the database without any personalization. Therefore, vanilla GANs are neither personalized nor zero-shot, making direct and fair comparisons with the methods

in this thesis difficult. Nonetheless, a wide array of work exists on vanilla GANs [104, 39, 40, 41, 42], primarily to address the data imbalance problem inherent in most ECG datasets rather than to perform zero-shot arrhythmia detection. **Personalized GANs** are similar to vanilla GANs, except they go a step further using transfer learning and fine-tuning to personalize the GAN to a certain patient before classification, such as in [43]. In Chapter 4, we compare our work with the global GAN synthesizer and end-to-end classifier proposed by Shaker *et al.* in [104]. Despite the limitations of our experimental setup, the results showcase the strength of the proposed methodology in this thesis.

**Personalized and zero-shot GANs** are fair to compare with this work, as proposed by Zhou *et al.* in [105]. First, the authors train a global generator that produces a 2-D coupling matrix similar to [100]. At the same time, a discriminator predicts the beat class from the coupling matrix. After the initial training, the discriminator is fine-tuned for each patient using only their healthy beats and any generated synthetic abnormal beats by the GAN. This fine-tuned discriminator is then used for arrhythmia detection. We compare our work with [105] later in Chapter 4.

### 3 Methods

#### 3.1 Dictionary Learning for the Healthy ECG Space

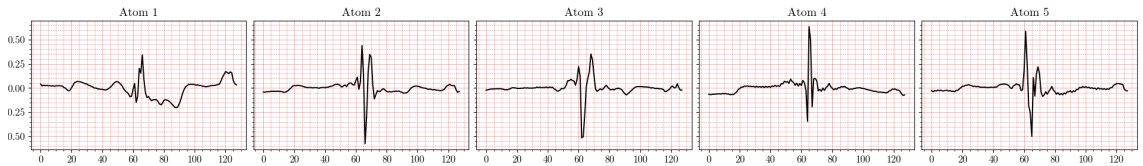
As in [89], the work in this thesis begins by building a dictionary to represent an individual’s healthy ECG beat space. The first task is to collect a small set of healthy beats,  $\mathbf{H}_p \in \mathbb{R}^{L \times B}$ , for an individual  $p$  (called the dictionary beat set of  $p$ ), following the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI) [106] (the recommendations are detailed in Chapter 4). Then, we use the Lasso formulation of 2.9 to simultaneously learn a sparse coefficient matrix,  $\mathbf{X}_p \in \mathbb{R}^{K \times B}$ , and a dictionary,  $\mathbf{D}_p \in \mathbb{R}^{L \times K}$ , for the healthy ECG beat space for  $p$ , where  $L$  is the length of each signal,  $K$  is the number of atoms in the dictionary, and  $B$  is the number of beats used for learning. The problem is formulated as follows:

$$\min_{\mathbf{D}_p, \mathbf{X}_p} \|\mathbf{H}_p - \mathbf{D}_p \mathbf{X}_p\|_2 + \lambda \|\mathbf{X}_p\|_1. \quad (3.1)$$

In order to solve 3.1, we implement a variant of the MOD [63]. As in the original MOD, we iteratively solve for  $\mathbf{D}_p^{\{t\}}$  and  $\mathbf{X}_p^{\{t\}}$  at each iteration  $t$  in an alternating fashion. Once again, the optimal  $\mathbf{D}_p^{\{t\}}$  at each iteration is given by  $\mathbf{D}_p^{\{t\}} = \mathbf{H}_p \mathbf{X}_p^{\{t\}+}$ . To find the optimal  $\mathbf{X}_p^{\{t\}}$ , we implement *the Alternating Direction Method of Multipliers* (ADMM) [107] similar to [90]. ADMM is a well-studied *split optimization algorithm* that can be applied to any optimization problem of the form:

$$\min_{\mathbf{x}} F_1(\mathbf{x}) + F_2(\mathbf{x}) \quad \text{subject to } \mathbf{x} \in \text{convex set } \mathbf{K}, \quad (3.2)$$

where  $F_1$  involves an  $\ell^1$  norm and  $F_2$  involves an  $\ell^2$  norm [108]. Thus, ADMM suits perfectly when solving for  $\mathbf{X}_p$  in the optimization problem 3.1. The formal implementation of the algorithm is provided in Algorithm 1. In Figure 3.1, we provide the first five atoms from the learned dictionary of patient 100 in the MIT-BIH database. A sparse combination of these atoms can form any healthy ECG in  $\mathbf{H}_p$  with a small representation error.



**Figure 3.1** The first five atoms of the dictionary for patient (a) in Figure 2.4. The dictionary is learned with Algorithm 1, where  $L = 128$ ,  $K = 20$ , and  $\lambda = 0.01$ .

---

**Algorithm 1** MOD Variant for the Lasso Formulation
 

---

```

1: procedure MOD_LASSO( $\mathbf{H}_p \in \mathbb{R}^{L \times B}$ ,  $K$ ,  $\lambda$ ,  $\tau_{cvg}$ , iterations)
2:    $\mathbf{D}_p \leftarrow \mathcal{N}(\mu = 0, \sigma^2 = 1)_{L \times K}$ 
3:    $\mathbf{D}_p \leftarrow \mathbf{D}_p \cdot \text{diag}(1 / \|\mathbf{D}_p\|_2)$  ▷ Normalize Atoms of  $\mathbf{D}_p$ 
4:   for  $it \leftarrow 1$  to iterations do
5:      $\mathbf{X}_p \leftarrow \text{ADMM\_LASSO}(\mathbf{D}_p, \mathbf{H}_p, \lambda)$ 
6:      $\mathbf{D}_p \leftarrow \mathbf{H}_p \mathbf{X}_p^+$ 
7:      $\mathbf{D}_p \leftarrow \mathbf{D}_p \cdot \text{diag}(1 / \|\mathbf{D}_p\|_2)$  ▷ Normalize Atoms of  $\mathbf{D}_p$ 
8:      $\tilde{e}_{\ell^1} \leftarrow \|\mathbf{H}_p - \mathbf{D}_p \mathbf{X}_p\|_2$ 
9:     if  $\tilde{e}_{\ell^1} < \tau_{cvg}$  then
10:       break
11:     end if
12:   end for
13:   return  $\mathbf{D}_p, \mathbf{X}_p$ 
14: end procedure

```

---

In conventional SRC, the dictionaries are overcomplete in order to represent a broader range of signals with a small number of atoms. The dictionaries for personalized arrhythmia detection are undercomplete ( $K < L$ ) as they should represent only a single class—the healthy ECG beats for patient  $p$ —and should not be able to represent the anomalous ECG beats. To this end, the dictionaries are learned to be as compact as possible, and in Section 3.4, we show that the dictionary for one individual fails to represent even the *healthy* ECG beats from *another* individual due to their morphological differences.

Finally, we note that dictionary learning is an offline procedure, meaning that the dictionaries for each individual  $p \in [1 \dots M]$  can be learned beforehand and stored. Thus, we form a collection of dictionaries for various individuals and keep expanding this collection for each new individual.

### 3.2 Representation Error-based Arrhythmia Detection

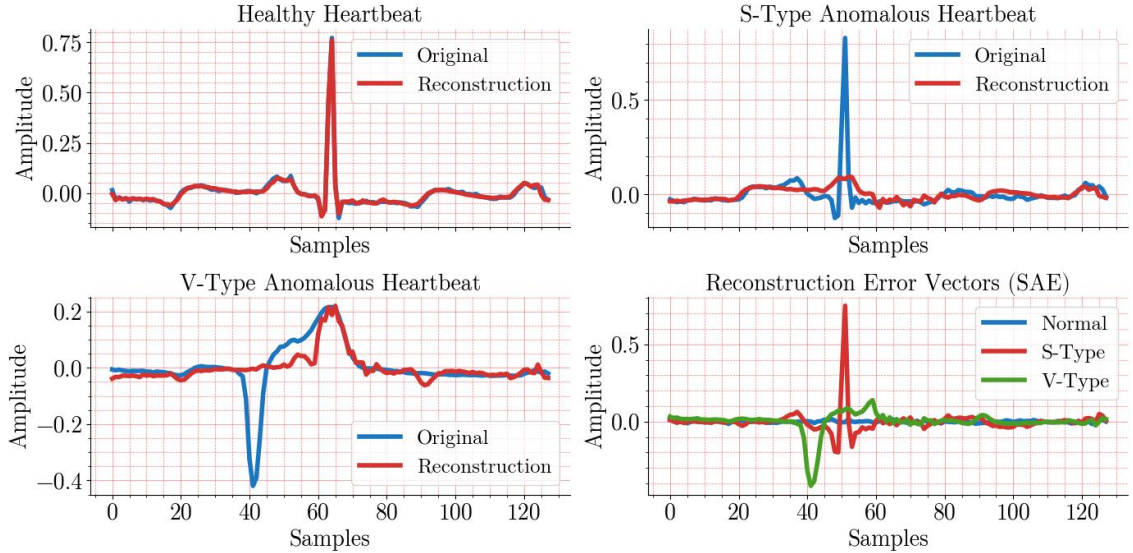
After learning the dictionary  $\mathbf{D}_p$  for the individual  $p$ , we can calculate the representation error of any ECG beat belonging to  $p$ . In [89], Carrera *et al.* use the iterative OMP algorithm to solve for the sparse coefficients and use Equation 2.15 to compute the residual error of a given signal. We start with the Lasso formulation of 2.7 for a given signal  $\mathbf{s}_p^{(i)}$  and  $\mathbf{D}_p$ :

$$\min_{\mathbf{x}_p^{(i)}} \|\mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{x}_p^{(i)}\|_2 + \lambda \|\mathbf{x}_p^{(i)}\|_1. \quad (3.3)$$

This optimization problem is equivalent to the update step for  $\mathbf{X}_p$  when learning the dictionary in Section 3.1. Therefore, we once again use the ADMM algorithm to find the sparse coefficient vector  $\mathbf{x}_p^{(i)}$ . After solving for  $\mathbf{x}_p^{(i)}$ , we can compute the sparse approximation error vector as:

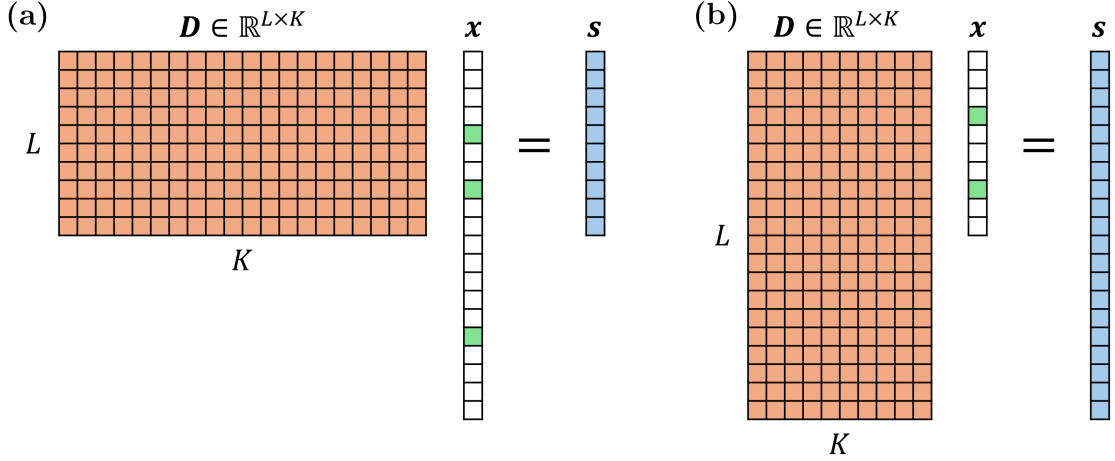
$$\tilde{\mathbf{e}}_{\ell^1} = \mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{x}_p^{(i)}. \quad (3.4)$$

Then, an empirical threshold can be selected as in [89] to classify any ECG belonging to  $p$  based on the energy of SAE, i.e.,  $\|\tilde{\mathbf{e}}_{\ell^1}\|_2^2$ . Figure 3.2 clarifies how this threshold mechanism can classify healthy beats from arrhythmia. The healthy beat is nearly perfectly reconstructed using the dictionary, whereas the anomalous beats cannot be reconstructed and produce large error vectors.



**Figure 3.2** Example of healthy, S-type, and V-type beats and their corresponding reconstructions. The bottom right plot shows the reconstruction error vectors for all heartbeat types. Both abnormal beats have noticeably large SAE compared to the healthy beat. The beats belong to patient 100 from the MIT-BIH database [48].

Much like OMP, ADMM is also an iterative algorithm and requires considerable floating point operations (FLOPs) to compute. Moreover, this process must be repeated for every new signal in an online monitoring setup. Having laid the groundwork, we pursue a different and more efficient approach to find the representation error by utilizing the fact that  $\mathbf{D}_p$  is an undercomplete dictionary ( $K < L$ ), as shown in Figure 3.3 (b). Since  $\mathbf{D}_p$  is undercomplete, a given healthy ECG signal may not be in the column space of  $\mathbf{D}_p$ , i.e.,  $\mathbf{s}_p^{(i)} \notin C(\mathbf{D}_p)$ . In such cases, it is impossible to fully represent the signal without any error:



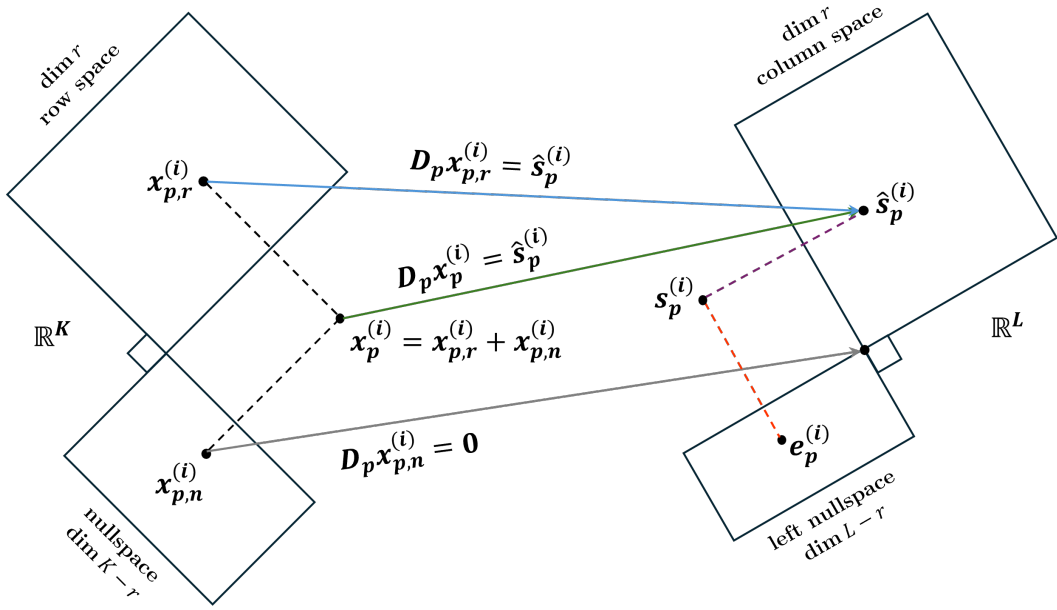
**Figure 3.3** Overcomplete (a) vs. undercomplete (b) dictionaries for the sparse representation of a signal,  $\mathbf{D}\mathbf{x} = \mathbf{s}$ .

$$\mathbf{s}_p^{(i)} = \mathbf{D}_p \mathbf{x}_p^{(i)} + \mathbf{e}_p^{(i)} \quad (3.5)$$

Moreover, the *Fundamental Theorem of Linear Algebra* [51] tells us that the left nullspace of  $\mathbf{D}_p$  (denoted as  $N(\mathbf{D}_p^T)$ ) is non-trivial and at least  $L - K$  dimensional, as depicted in Figure 3.4. The figure helps us paint a geometric view of the whole picture. One way to describe the sparse representation of  $\mathbf{s}_p^{(i)}$  is that the coefficients  $\mathbf{x}_p^{(i)}$  define how the column vectors of  $\mathbf{D}_p$  should be combined to construct the best approximation of  $\mathbf{s}_p^{(i)}$ . This is the standard view of matrix-vector multiplication, i.e.,  $\mathbf{D}_p \mathbf{x}_p^{(i)}$ . Another equivalent description is the one painted by Figure 3.4; the transformation defined by  $\mathbf{D}_p$  takes any coefficient vector  $\mathbf{x}_{p,r}^{(i)}$  from the row space  $C(\mathbf{D}_p^T)$  to a unique signal  $\hat{\mathbf{s}}_p^{(i)}$  in the column space  $C(\mathbf{D}_p)$ . Since  $\mathbf{D}_p$  is undercomplete ( $K < L$ ), the column vectors cannot span the entire signal space  $\mathbb{R}^L$ , but can only span a  $rank(\mathbf{D}_p)$  dimensional subspace (i.e., the column space, which is at most  $K$  dimensional). The part of the signal that does not lie in  $C(\mathbf{D}_p)$ , the error  $\mathbf{e}_p^{(i)}$ , cannot be represented by  $\mathbf{D}_p$ . Thus the error lies in the left nullspace,  $N(\mathbf{D}_p^T)$ , which is perpendicular to  $C(\mathbf{D}_p)$ . In the following sections, we describe how to utilize the left nullspace of  $\mathbf{D}_p$  to significantly reduce the computational complexity of the classification task by eliminating the need to compute the coefficient vector and directly inferring the error vector.

### 3.2.1 Nullspace Projection Error

When the dictionary is undercomplete, the left nullspace of  $\mathbf{D}_p$ ,  $N(\mathbf{D}_p^T)$ , is non-trivial, and the representation error resides in the left nullspace. Let  $\mathbf{N}_p$  be the matrix whose columns are the orthonormal basis vectors of  $N(\mathbf{D}_p^T)$ , i.e.,  $\mathbf{D}_p^T \mathbf{N}_p = \mathbf{0}$ . In other words, multiplying  $\mathbf{N}_p$  or any vector spanned by the columns of  $\mathbf{N}_p$  with



**Figure 3.4** [51]: The four fundamental subspaces and the action of  $\mathbf{D}_p$  with rank  $r$ .  $\mathbf{D}_p$  takes the row space to the column space (blue arrow) and the nullspace to zero (gray arrow). The signal  $\mathbf{s}_p^{(i)}$  may not lie in the column space, in which case the representation error emerges and resides in the left nullspace (dotted red line). The transformation,  $\mathbf{D}_p \mathbf{x}_p^{(i)}$  (green arrow), can only represent  $\hat{\mathbf{s}}_p^{(i)}$ , which is the component of  $\mathbf{s}_p^{(i)}$  in the column space (dotted purple line). If  $\mathbf{D}_p$  is undercomplete and full rank ( $r = K$ ), there is a unique solution  $\mathbf{x}_p^{(j)}$  for any  $\hat{\mathbf{s}}_p^{(j)}$  in the column space, as the nullspace is trivial and only contains the zero vector; in other words  $\mathbf{x}_p^{(i)} = \mathbf{x}_{p,r}^{(i)}$  and  $\mathbf{x}_{p,n}^{(i)} = \mathbf{0}$ .

$\mathbf{D}_p^T$  equals zero. Then, we define  $\mathbf{F}_p \in \mathbb{R}^{L-K \times L}$  to be the transpose of  $\mathbf{N}_p$ , and thus  $\mathbf{F}_p$  is the left annihilator matrix of  $\mathbf{D}_p$ , i.e.:

$$\mathbf{D}_p^T \mathbf{N}_p = (\mathbf{D}_p^T \mathbf{N}_p)^T = \mathbf{N}_p^T \mathbf{D}_p = \mathbf{F}_p \mathbf{D}_p = \mathbf{0}. \quad (3.6)$$

When a new signal  $\mathbf{s}_p^{(i)}$  belonging to  $p$  is collected, we can calculate the alignment of  $\mathbf{s}_p^{(i)}$  and the left nullspace  $N(\mathbf{D}_p^T)$  by multiplying  $\mathbf{F}_p$  and  $\mathbf{s}_p^{(i)}$ , i.e.,  $\mathbf{F}_p \mathbf{s}_p^{(i)}$ . By multiplying Equation 3.5 from the left with  $\mathbf{F}_p$ , we can compute this alignment:

$$\begin{aligned} \mathbf{F}_p \mathbf{s}_p^{(i)} &= \overbrace{\mathbf{F}_p \mathbf{D}_p \mathbf{x}_p^{(i)}}^{\mathbf{F}_p \mathbf{D}_p = \mathbf{0}} + \mathbf{F}_p \mathbf{e}_p^{(i)} \\ \mathbf{F}_p \mathbf{s}_p^{(i)} &= \mathbf{F}_p \mathbf{e}_p^{(i)} = \tilde{\mathbf{e}}_N. \end{aligned} \quad (3.7)$$

We call the resulting vector,  $\tilde{\mathbf{e}}_N$ , the **nullspace projection error (NPE)**. Multiplication with  $\mathbf{F}_p$  effectively removes the healthy beat components from  $\mathbf{s}_p^{(i)}$  through a simple matrix-vector multiplication. Therefore, if  $\mathbf{s}_p^{(i)}$  is a healthy ECG beat, its NPE would be small; however, if  $\mathbf{s}_p^{(i)}$  is anomalous, then its NPE would be much

greater in magnitude. In Chapter 4, we show that nullspace projection error and sparse approximation error perform equally. Yet, SAE requires the sparse coefficients to be computed with an iterative algorithm either by solving 2.7 (e.g., through OMP) or 3.3 (e.g., through ADMM). On the other hand, NPE can be calculated directly through the input signal with a single matrix-vector multiplication, making it much more computationally efficient than SAE. Note that  $\mathbf{F}_p$  can be calculated once and stored along with  $\mathbf{D}_p$  and accessed when required, thus finding  $\mathbf{F}_p$  does not affect the computational cost of NPE.

### 3.2.2 Least Squares Approximation Error

Additionally, we provide another computationally efficient method as an alternative to NPE. As discussed in Section 2.2.4, the authors of [86] proved that in many multi-class classification problems, CRC performs similarly to SRC. As opposed to SRC, CRC uses  $\ell^2$  regularized coefficients instead of  $\ell^1$ . We adopt a similar strategy and rewrite the Lasso formulation in 3.3 as the Ridge formulation:

$$\min_{\mathbf{x}_p^{(i)}} \|\mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{x}_p^{(i)}\|_2 + \lambda \|\mathbf{x}_p^{(i)}\|_2. \quad (3.8)$$

The Ridge formulation can be solved directly using the method of regularized least squares:

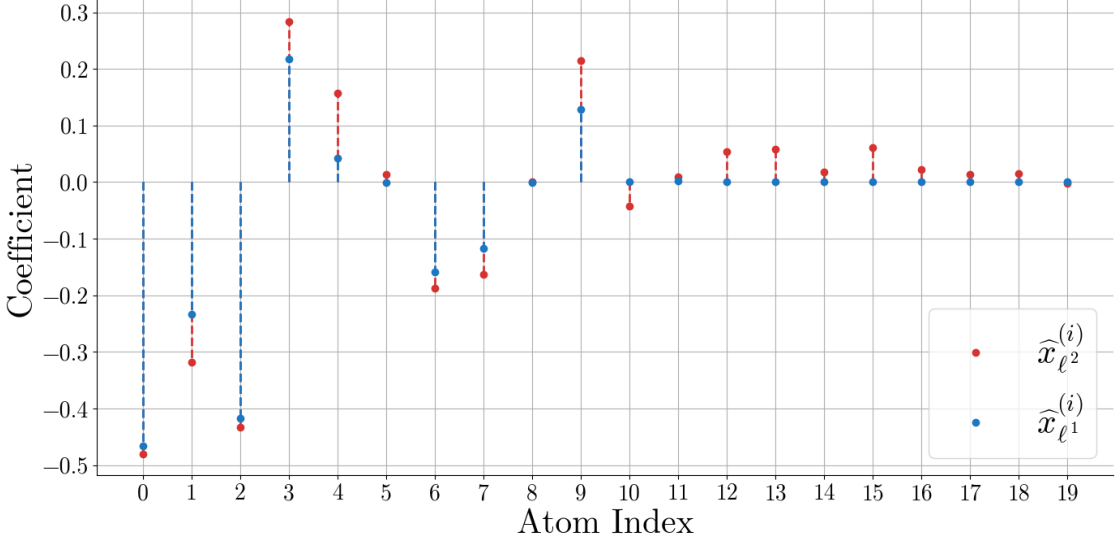
$$\mathbf{x}_{\ell^2}^{(i)} = \underbrace{(\mathbf{D}_p^T \mathbf{D}_p + \lambda \mathbf{I})^{-1} \mathbf{D}_p^T}_{\mathbf{L}_p} \mathbf{s}_p^{(i)}. \quad (3.9)$$

When  $\lambda = 0$ , Equation 3.9 becomes the ordinary least squares solution. However, ordinary least squares cannot provide a solution if  $\mathbf{D}_p$  is not full rank, as  $\mathbf{D}_p^T \mathbf{D}_p$  has no inverse. Moreover, even if a solution exists, it may be unstable if  $\mathbf{D}_p$  is ill-conditioned (i.e., full rank, but one or a few singular values are too small). Therefore, we prefer the Ridge formulation, which ensures that the solution is well-posed; when  $\lambda > 0$ , it controls the trade-off between smaller coefficients and the magnitude of the representation error.

After finding the coefficients  $\mathbf{x}_{\ell^2}^{(i)}$ , the representation error vector can be calculated simply as  $\tilde{\mathbf{e}}_{\ell^2} = \mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{x}_{\ell^2}^{(i)}$ . We call the resulting vector,  $\tilde{\mathbf{e}}_{\ell^2}$ , the **least squares approximation error (LAE)**. Similar to  $\mathbf{F}_p$ ,  $\mathbf{L}_p$  in Equation 3.9 can be calculated and stored; therefore LAE can be equivalently expressed as:

$$\tilde{\mathbf{e}}_{\ell^2} = \mathbf{s}_p^{(i)} - \mathbf{D}_p \mathbf{L}_p \mathbf{s}_p^{(i)}. \quad (3.10)$$

In Figure 3.5, we provide an example pair of coefficient vectors  $\mathbf{x}_{\ell^1}^{(i)}$  and  $\mathbf{x}_{\ell^2}^{(i)}$ . Although the Lasso formulation 3.3 provides sparser solutions, the energy of the



**Figure 3.5** Example coefficient vectors  $\mathbf{x}_{\ell^1}^{(i)}$  and  $\mathbf{x}_{\ell^2}^{(i)}$  for a healthy ECG beat using a dictionary with  $K = 20$  atoms.  $\mathbf{x}_{\ell^1}^{(i)}$  is more sparse compared to  $\mathbf{x}_{\ell^2}^{(i)}$ .

representation errors (SAE and LAE) are almost identical for our task.

In the next section, we provide the number of FLOPs needed to calculate NPE, LAE, and SAE and show that the former two require significantly less computational power compared to SAE. On top of the computational efficiency, all of the approximation errors yield almost identical classification performance with a simple thresholding-based classifier as in [89], which we discuss in Chapter 4.

### 3.2.3 Computational Complexity Analysis

We begin the analysis with NPE. Given the pre-constructed left annihilator matrix  $\mathbf{F}_p \in \mathbb{R}^{L-K \times L}$  and an ECG beat  $\mathbf{s}_p^{(i)} \in \mathbb{R}^L$  for individual  $p$ , the NPE is simply the result of a matrix-vector multiplication. Therefore, the exact number of FLOPs for calculating NPE is:

$$T_{\text{NPE}}(L, K) = 2 \cdot L \cdot (L - K) \text{ FLOPs.} \quad (3.11)$$

The analysis for LAE is similar. Once again we have a pre-constructed matrix  $\mathbf{L}_p \in \mathbb{R}^{K \times L}$  and an ECG beat  $\mathbf{s}_p^{(i)} \in \mathbb{R}^L$ . Following Equation 3.10, LAE can be calculated with a matrix-vector multiplication of  $\mathbf{L}_p$  and  $\mathbf{s}_p^{(i)}$ , followed by a matrix-vector multiplication of  $\mathbf{D}_p \in \mathbb{R}^{L \times K}$  and the intermediate result, and lastly a subtraction from the original signal  $\mathbf{s}_p^{(i)}$ . Thus, the number of FLOPs needed to calculate LAE is:

$$T_{\text{LAE}}(L, K) = (4 \cdot K + 1) \cdot L \text{ FLOPs.} \quad (3.12)$$

**Table 3.1** Number of FLOPs needed for calculating different representation errors and the processing time of a single ECG beat on an i7-10870H CPU and an RTX 3080 Laptop GPU. The measurements are averaged over 10 runs and 1000 beats.

Error	Complexity (FLOPs)	CPU Time ( $\mu\text{s}$ )	GPU Time ( $\mu\text{s}$ )
SAE	$T_{\text{OMP}}(L, K, \kappa) + (2 \cdot K + 1) \cdot L$	4.7933	11.6808
NPE	$2 \cdot L \cdot (L - K)$	0.2047	0.1032
LAE	(i) $2 \cdot L^2$	(i) 0.2498	(i) 0.1062
	(ii) $(4 \cdot K + 1) \cdot L$	(ii) 0.2133	(ii) 0.1196
ARE	$(4 \cdot K + 1) \cdot L$	0.3522	0.3164
ARE-R	$(2 \cdot K + 1) \cdot 2 \cdot L$	0.3558	0.3096

Another approach for calculating LAE is to parenthesize Equation 3.10 as  $(\mathbf{I} - \mathbf{D}_p \mathbf{L}_p) \mathbf{s}_p^{(i)}$  and pre-construct the matrix  $\mathbf{I} - \mathbf{D}_p \mathbf{L}_p$ . Then LAE can be calculated with a single matrix-vector multiplication, and the required FLOPs would be:

$$T_{\text{LAE}}(L) = 2 \cdot L^2. \quad (3.13)$$

The more preferable approach depends on  $L$  and  $K$ , and the computational platform. In general, Equation 3.12 will require fewer FLOPs since  $K < L$ ; however, Equation 3.13 may be faster in parallel platforms since all the inner products can be computed in parallel.

The analysis for SAE requires two parts. First, we find the sparse coefficients  $\mathbf{x}_p^{(i)}$  with an algorithm described in Section 2.2.2 or 3.2. Then, we compute the error using the coefficients and the dictionary as in Equation 2.15 or 3.4. Since there are multiple ways to find the sparse coefficients, there is not an exact number of FLOPs to calculate SAE. For analysis, we select OMP as the algorithm to find the sparse coefficients similar to Carrera *et al.* [89]. In OMP, we have one more hyperparameter: the maximum number of non-zero coefficients,  $\kappa$ . The number of FLOPs required for OMP is approximately:

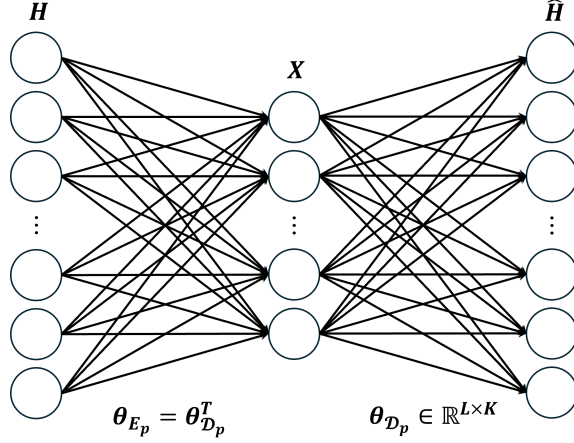
$$T_{\text{OMP}}(L, K, \kappa) \approx 2 \cdot L \cdot \kappa \cdot (\kappa + 1.5) + 2 \cdot \kappa \cdot K \cdot (L + 1) \text{ FLOPs.} \quad (3.14)$$

With the sparse codes, SAE can be calculated with an additional  $(2 \cdot K + 1) \cdot L$  FLOPs.

### 3.3 Joint Embedding and Dictionary Learning via Personalized Autoencoders

In this section, we provide an autoencoder-based dictionary learning method as an improvement over MOD-based dictionary learning. In Chapter 4, we show that the

representation errors obtained from autoencoder-based dictionaries provide better classification performance than all three error types, SAE, NPE, and LAE when used in a simple thresholding-based classifier.



**Figure 3.6** Symmetrical single hidden-layer autoencoder network architecture for dictionary learning. After training, the learned model weights  $\theta_{\mathcal{D}_p}$  correspond to the dictionary.

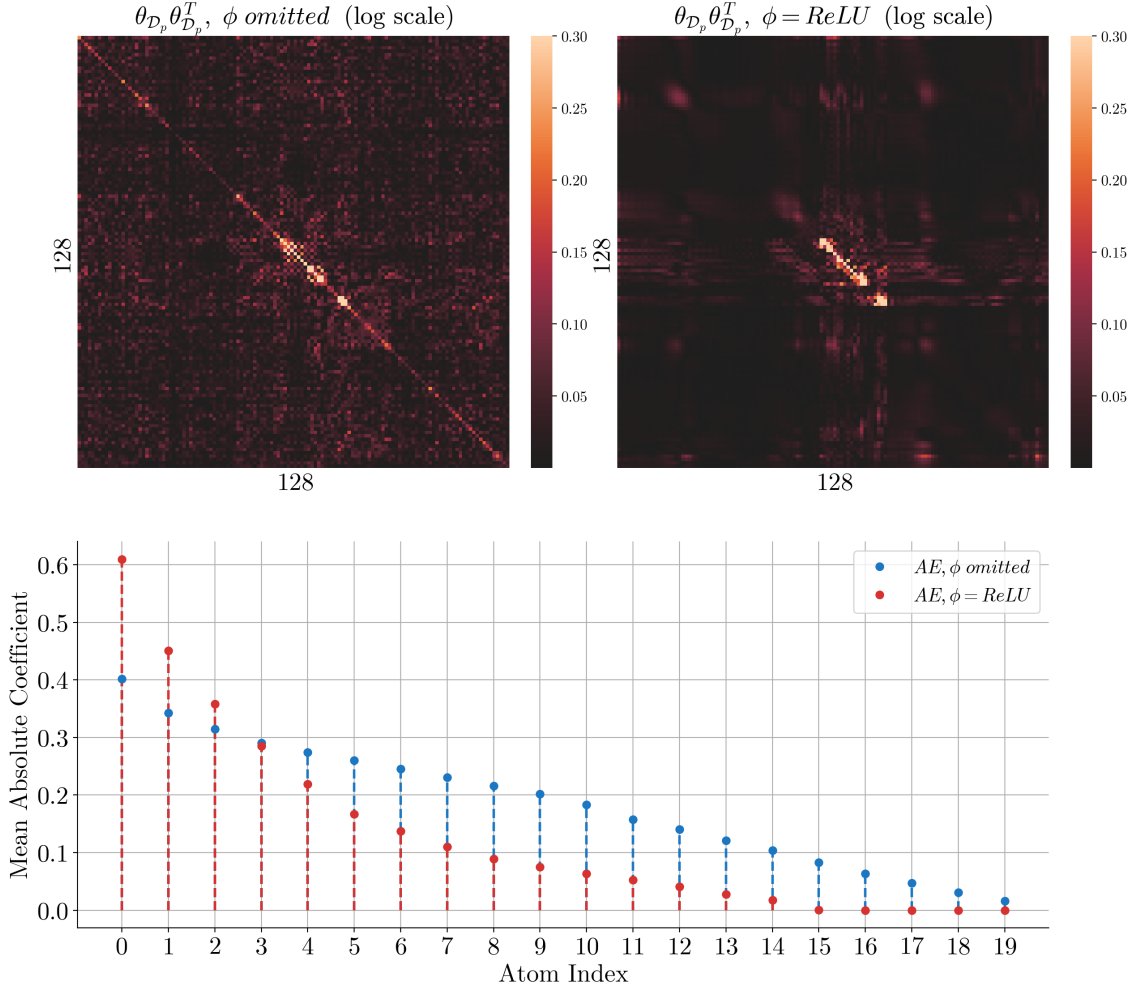
We follow the same procedure as in Section 3.1 to collect a small set of healthy beats,  $\mathbf{H}_p \in \mathbb{R}^{L \times B}$ , for an individual  $p$ . Then, we train a symmetrical single hidden-layer autoencoder (AE), whose decoder,  $\theta_{\mathcal{D}_p} \in \mathbb{R}^{L \times K}$ , corresponds to the dictionary. In a symmetrical autoencoder, the encoder is tied to the decoder, i.e.,  $\theta_{E_p} = \theta_{\mathcal{D}_p}^T$ . This network architecture is illustrated in Figure 3.6. The autoencoder is trained by minimizing the following optimization problem through gradient descent:

$$\hat{\theta}_{\mathcal{D}_p} = \arg \min_{\theta_{\mathcal{D}_p}} \left\| \mathbf{H}_p - \theta_{\mathcal{D}_p} \phi(\theta_{\mathcal{D}_p}^T \mathbf{H}_p) \right\|_2 + \left\| \theta_{\mathcal{D}_p}^T \mathbf{H}_p \right\|_1. \quad (3.15)$$

There is no need to regularize  $\theta_{\mathcal{D}_p}$  since keeping the decoder and encoder tied implicitly forces the columns of  $\theta_{\mathcal{D}_p}$  to be orthonormal, i.e.,  $\left\| \theta_{\mathcal{D}_p} \theta_{\mathcal{D}_p}^T - \mathbf{I} \right\|_2 \approx \mathbf{0}$ . We train two such autoencoders; the first one omits  $\phi$ , and thus, it learns a linear model as in 3.1. The second one sets  $\phi = ReLU$  for non-linearity, which additionally acts as a regularizer for the activations (i.e., the coefficients).

As opposed to the sparse representation of a new signal,  $\mathbf{s}_p^{(i)}$ , there is no need to solve for the coefficient vector since the encoding stage of the autoencoder already computes it, i.e.,  $\mathbf{x}_p^{(i)} = \theta_{\mathcal{D}_p}^T \mathbf{s}_p^{(i)}$ . Thus, calculating the representation error from the reconstructions of the autoencoder is computationally more efficient than SAE. In Figure 3.7, we provide an example set of activations from both autoencoders and show that when  $\phi = ReLU$ , the coefficients become sparser than when  $\phi$  is omitted. On the upper side of the figure, we plot the entire transformation  $\theta_{\mathcal{D}_p} \theta_{\mathcal{D}_p}^T$  for both autoencoders. For the first autoencoder where  $\phi$  is omitted, the transformation affects the entire beat with an emphasis on the QRS complex. For the second

autoencoder where  $\phi = \text{ReLU}$ , the transformation can focus entirely on the localized region around the QRS complex due to the non-linearity.



**Figure 3.7** Example autoencoder weight matrices,  $\theta_{\mathcal{D}_p} \theta_{\mathcal{D}_p}^T$ , acting on  $\mathbf{H}_p$  for the two autoencoders. On the left,  $\theta_{\mathcal{D}_p} \theta_{\mathcal{D}_p}^T$  is without any activation functions, and on the right,  $\theta_{\mathcal{D}_p} \theta_{\mathcal{D}_p}^T$  has a ReLU activation in-between. The bright spots on the diagonal on both matrices correspond to the QRS complex, typically the largest and most prominent wave in the ECG beat. The bottom plot shows the sparsifying effect of ReLU on the coefficients,  $\mathbf{X}_p = \theta_{\mathcal{D}_p} \mathbf{H}_p$ , where the coefficients decay faster when ReLU is present (i.e., sparser, in red), and much slower when it is not (in blue).

### 3.3.1 Autoencoder Reconstruction Error

After training the two autoencoders, we can obtain the reconstruction (i.e., representation) error for a given signal  $\mathbf{s}_p^{(i)}$  simply as:

$$\tilde{\mathbf{e}}_{AE} = \mathbf{s}_p^{(i)} - \theta_{\mathcal{D}_p} \phi(\theta_{\mathcal{D}_p}^T \mathbf{s}_p^{(i)}). \quad (3.16)$$

We call the reconstruction error vector,  $\tilde{\mathbf{e}}_{\mathbf{AE}}$ , the **autoencoder reconstruction error (ARE)**. When  $\phi$  is omitted, we can see from Equations 3.10 and 3.16 that calculations for the LAE and ARE are identical with matrices of the same size. Thus, their computational complexity in terms of FLOPs is identical:

$$T_{\text{ARE}}(L, K) = T_{\text{LAE}}(L, K) = (4 \cdot K + 1) \cdot L \text{ FLOPs.} \quad (3.17)$$

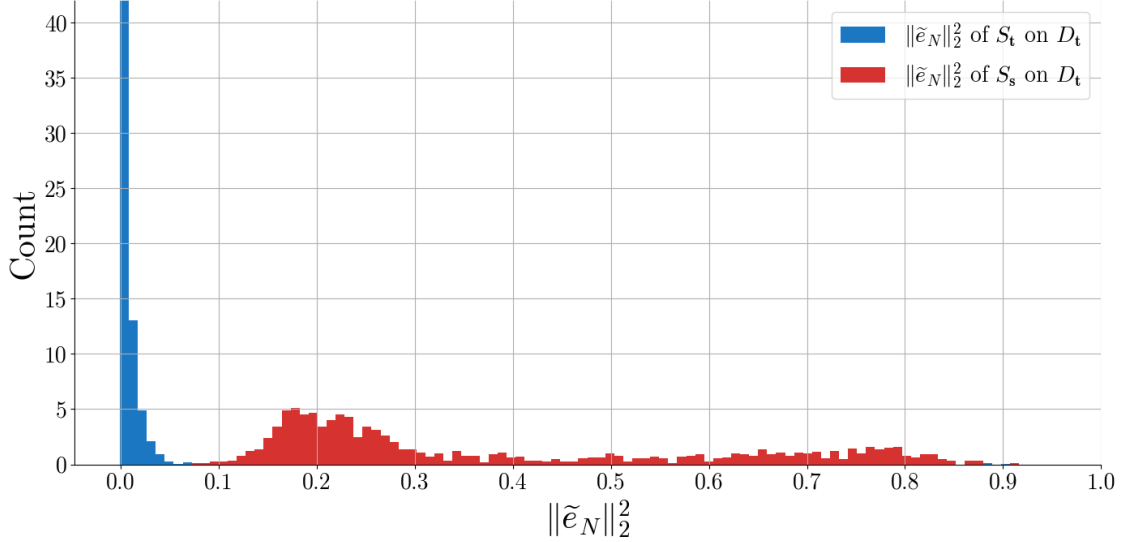
When  $\phi = \text{ReLU}$ , the only difference is an additional  $L$  FLOPs to calculate the activations. In Table 3.1, We provide the overall complexity analysis for all four error vectors, SAE, NPE, LAE, and ARE, along with their runtime performance measured on a CPU and a GPU. Although their computational complexity is identical, due to differences in the programming frameworks, LAE and ARE perform differently on our testing hardware. Nonetheless, among all four errors, the NPE achieves the lowest computational complexity and the fastest runtime, especially when parallel processing units are available. Later in Chapter 4, we compare the performances of four different thresholding-based classifiers built on the respective error vectors and provide their classification performance.

### 3.4 Sparse Representation-based Domain Adaptation

In this section of the thesis, we aim to tackle two important questions in the literature for personalized and zero-shot ECG arrhythmia detection: (i) How can we train robust machine learning models to perform arrhythmia detection for healthy individuals with no anomalous ECG to learn from? (ii) How can we utilize the abundant normal and anomalous ECG beats from existing patients to aid in arrhythmia detection for healthy individuals? To address both problems simultaneously, we draw inspiration from the related literature work [44], in which the authors transform the normal heartbeats of a patient into potential anomalous beats through an LTI degradation system by modeling the relationship between normal and anomalous beats. Differently, in this thesis, we investigate the morphological relationship between the healthy ECG beats of two individuals named *source*,  $\mathbf{s}$ , and *target*,  $\mathbf{t}$ , and build a monitoring system to detect anomalous beats. In our study,  $\mathbf{s}$  is a patient registered to the system with a certain number of normal and anomalous beats. At the same time,  $\mathbf{t}$  is a healthy individual for whom we only have access to a small number of healthy beats but no abnormal beats.

Following either Section 3.1 or 3.3, we can learn a dictionary,  $\mathbf{D}_{\mathbf{t}}$ , to represent the healthy ECG beat space of  $\mathbf{t}$  using the small number of healthy beats. Using  $\mathbf{D}_{\mathbf{t}}$ , we calculate the NPE for both  $\mathbf{t}$ 's and  $\mathbf{s}$ 's beats. In Figure 3.8, we present these representation errors using two example patients from the MIT-BIH dataset corresponding to  $\mathbf{s}$  and  $\mathbf{t}$ . Despite being healthy beats, the beats of  $\mathbf{s}$  are not

represented well in  $\mathbf{D}_t$  due to morphological differences between the beats of  $\mathbf{s}$  and  $\mathbf{t}$ . In mathematical terms, the set of healthy beats of  $\mathbf{s}$ ,  $\mathbf{S}_s$ , are not in the column space of  $\mathbf{D}_t$ , i.e.,  $\mathbf{S}_s \notin C(\mathbf{D}_t)$ , and the representations result in large errors.



**Figure 3.8** Histogram of NPE energies,  $\|\tilde{e}_N\|_2^2$ , for the healthy beats of source,  $\mathbf{S}_s$ , and target,  $\mathbf{S}_t$ , on the dictionary of target,  $\mathbf{D}_t$ . Despite  $\mathbf{D}_t$  being learned on healthy ECG beats, it fails to represent the healthy beats of  $\mathbf{s}$  due to morphological differences between the two patients' ECGs.

Figure 3.8 makes it clear that we cannot directly use  $\mathbf{S}_s$  to aid in arrhythmia detection for  $\mathbf{t}$ . For a thresholding-based classifier, despite being healthy beats, all of  $\mathbf{S}_s$  may have been classified as arrhythmia due to their large representation errors. Therefore, we seek a transformation that adapts the beats of  $\mathbf{s}$  to be morphologically consistent with the beats of  $\mathbf{t}$ . We linearly model this transformation with a **morphology transformation matrix (MTM)**, denoted with  $\mathbf{Q}_{s \rightarrow t}$ , as follows:

$$\mathbf{S}_{s \rightarrow t} = \mathbf{Q}_{s \rightarrow t} \mathbf{S}_s. \quad (3.18)$$

The resulting beats,  $\mathbf{S}_{s \rightarrow t}$ , are the beats adapted from  $\mathbf{s}$  to  $\mathbf{t}$  to have a similar morphology with  $\mathbf{t}$ . Finding such a transformation from  $\mathbf{s}$  to  $\mathbf{t}$  can be challenging since only a very limited number of healthy beats are available for  $\mathbf{t}$ , and no anomalous beats. This may result in overfitting to this small number of beats for any learning algorithm. With such a problem at hand, we turn to sparse representation. SR allows us to generalize well even with limited data by capturing the most important features of  $\mathbf{t}$ 's healthy beats in a compact dictionary,  $\mathbf{D}_t$ . A well-learned dictionary can sparsely represent the healthy ECG beats of  $\mathbf{t}$  with minimal representation error. In Figure 3.9, we depict a geometric view of the information encoded within  $\mathbf{D}_t$ . The column space of  $\mathbf{D}_t$  is adjusted during learning to be a compact subspace that

can represent  $\mathbf{t}$ 's healthy beats,  $\mathbf{S}_t$ , with minimal error, i.e.  $\|\mathbf{S}_t - \mathbf{D}_t \mathbf{X}_t\|_2 \approx \mathbf{0}$ . In other words, if an ECG beat can be represented in  $\mathbf{D}_t$ , we can infer that it belongs to the healthy ECG domain of  $\mathbf{t}$ . As such, we form the following optimization problem to adapt  $\mathbf{s}$ 's beats to  $\mathbf{t}$ :

$$\hat{\mathbf{Q}}_{\mathbf{s} \rightarrow \mathbf{t}}, \hat{\mathbf{X}}_{\mathbf{s}^*} = \arg \min_{\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}, \mathbf{X}_{\mathbf{s}^*}} \|\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}} \mathbf{S}_s - \mathbf{D}_t \mathbf{X}_{\mathbf{s}^*}\|_2^2 + \lambda \|\mathbf{X}_{\mathbf{s}^*}\|_1 + \gamma \|\mathbf{S}_s - \mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}} \mathbf{S}_s\|_2^2, \quad (3.19)$$

where  $\lambda$  and  $\gamma$  are positive trade-off terms, and  $\mathbf{X}_{\mathbf{s}^*}$  is the sparse coefficient matrix for the domain-adapted ECG beats of  $\mathbf{s}$ . The goal of this optimization problem is to find an MTM such that the transformed beats,  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}$ , are represented well on the *target's* dictionary,  $\mathbf{D}_t$ . As before,  $\lambda$  determines the sparsity of the coefficients. The last term,  $\gamma \|\mathbf{S}_s - \mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}} \mathbf{S}_s\|_2^2$ , ensures that the problem does not have a trivial solution at  $\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}} = \mathbf{0}$ ,  $\mathbf{X}_{\mathbf{s}^*} = \mathbf{0}$ . Moreover, for large values of  $\gamma$ , it ensures that  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}$  does not deviate too much from  $\mathbf{S}_s$  and overfit onto  $\mathbf{D}_t$ .

In Figure 3.10, we provide a different view of what 3.19 aims to achieve. Before the transformation,  $\mathbf{s}$ 's healthy beats,  $\mathbf{S}_s$ , are not represented well on  $\mathbf{D}_t$  as shown in Figure 3.8. After the transformation, i.e.,  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}} = \mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}} \mathbf{S}_s$ , the resulting beats can be represented by  $\mathbf{D}_t$ . Since  $\mathbf{D}_t$  represents a compact subspace for the healthy beats of  $\mathbf{t}$ , any transformed beats,  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}$ , can also be classified as healthy beats for  $\mathbf{t}$ .

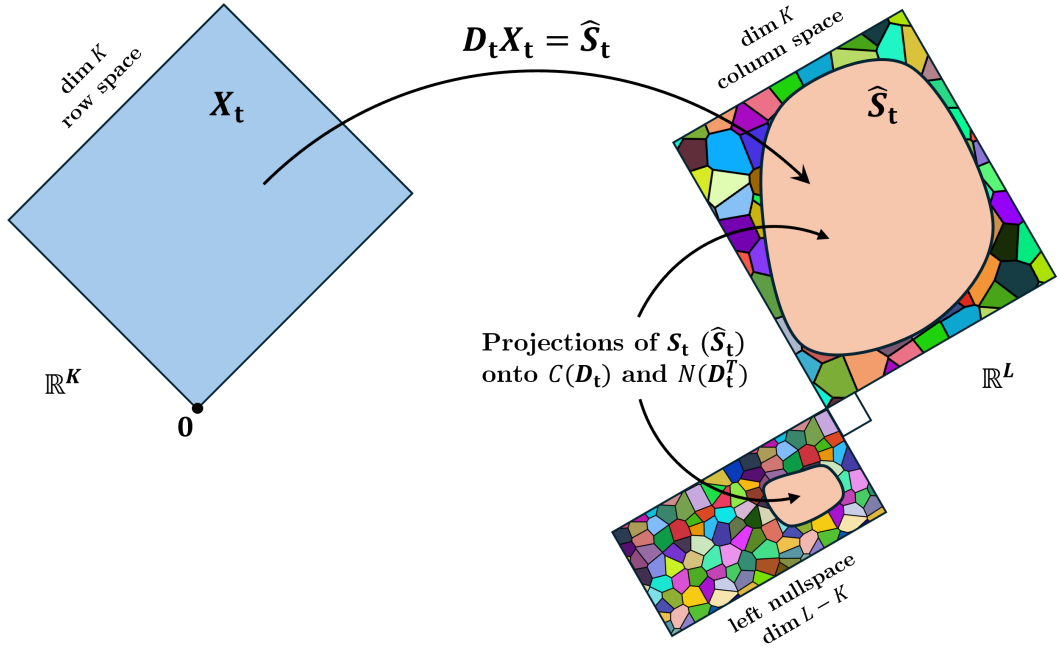
In order to solve 3.19, we follow a similar strategy to the MOD, and iteratively solve for  $\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}$  and  $\mathbf{X}_{\mathbf{s}^*}$  in an alternating fashion. Given  $\mathbf{D}_t$  and  $\mathbf{S}_s$  at each iteration, the solution for the morphology transformation matrix at the last iteration  $t - 1$ , denoted by  $\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t-1\}}$ , is fixed and the problem is solved for  $\mathbf{X}_{\mathbf{s}^*}^{\{t\}}$  via the reduced form of 3.19:

$$\hat{\mathbf{X}}_{\mathbf{s}^*}^{\{t\}} = \arg \min_{\mathbf{X}_{\mathbf{s}^*}^{\{t\}}} \left\| \mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t-1\}} - \mathbf{D}_t \mathbf{X}_{\mathbf{s}^*}^{\{t\}} \right\|_2^2 + \lambda \left\| \mathbf{X}_{\mathbf{s}^*}^{\{t\}} \right\|_1, \quad (3.20)$$

where  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t-1\}}$  are the transformed and unit normalized signals at step  $t - 1$  using  $\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t-1\}}$ . The optimization problem 3.20 is solved with ADMM as in Section 3.2. After obtaining the sparse coefficients for step  $t$ ,  $\mathbf{X}_{\mathbf{s}^*}^{\{t\}}$  is fixed and  $\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t\}}$  is obtained by solving:

$$\hat{\mathbf{Q}}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t\}} = \arg \min_{\mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t\}}} \left\| \mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t\}} \mathbf{S}_s - \mathbf{D}_t \mathbf{X}_{\mathbf{s}^*}^{\{t\}} \right\|_2^2 + \gamma \left\| \mathbf{S}_s - \mathbf{Q}_{\mathbf{s} \rightarrow \mathbf{t}}^{\{t\}} \mathbf{S}_s \right\|_2^2. \quad (3.21)$$

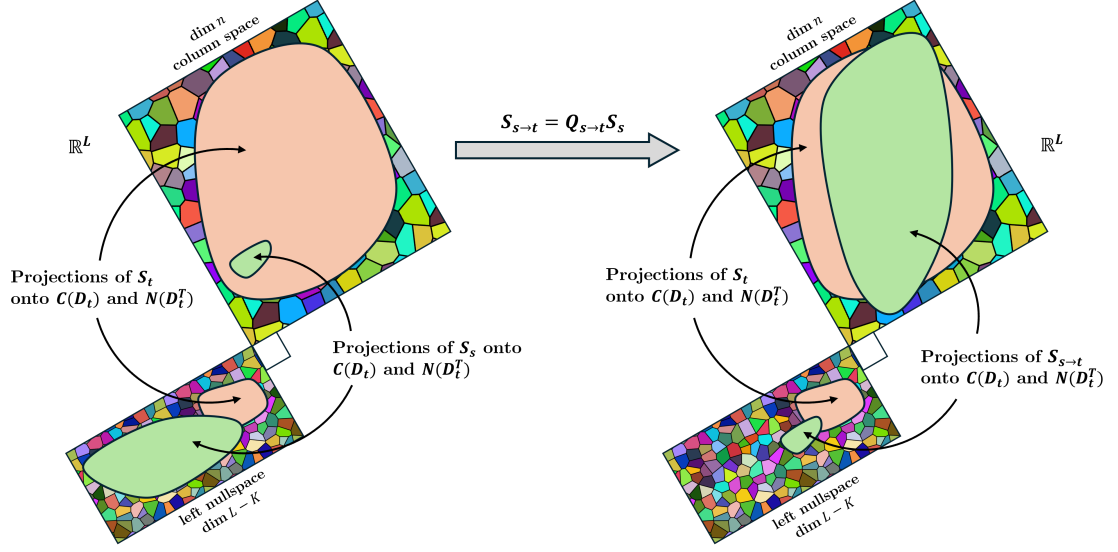
Despite being a quadratic function with a closed-form solution, we optimize 3.21 through gradient descent to obtain a smooth transition from  $\mathbf{S}_s$  to  $\mathbf{S}_{\mathbf{s} \rightarrow \mathbf{t}}$  in a fixed



**Figure 3.9** For a well-learned dictionary for  $\mathbf{t}$ , the column space,  $C(\mathbf{D}_t)$ , represents a compact subspace for the healthy ECG beats of  $\mathbf{t}$ . The projections of  $\mathbf{t}$ 's beats onto  $C(\mathbf{D}_t)$ , namely  $\hat{\mathbf{S}}_t$ , contain most of the components of  $\mathbf{S}_t$ , whereas the projections onto  $N(\mathbf{D}_t^T)$ —the residuals or representation errors—contain very few components, and have low signal energy, i.e.  $\|\mathbf{S}_t - \mathbf{D}_t \mathbf{X}_t\|_2^2 \approx \mathbf{0}$ . The many colors in  $C(\mathbf{D}_t)$  and  $N(\mathbf{D}_t^T)$  represent different types of signals that also fall onto these subspaces by coincidence. The key point of the figure is that most components of the signals in  $\mathbf{S}_t$  fall onto  $C(\mathbf{D}_t)$  rather than  $N(\mathbf{D}_t^T)$ .

number of steps. We provide a step-by-step implementation of the overall procedure to derive the morphology transformation matrix in Algorithm 2.

In Figure 3.11, we provide the representation errors of the transformed healthy ECG beats of  $\mathbf{s}$  on  $\mathbf{D}_t$ . After comparing with Figure 3.8, it is evident that the MTM adapts the morphology of  $\mathbf{S}_s$  to align with the healthy beats of  $\mathbf{t}$ , and improves their representation on  $\mathbf{D}_t$ . Moreover, experiments in the related literature work [44] show that there is a linear relationship between the healthy and anomalous beats of an individual. Therefore, our linear MTMs, which transform the healthy beats of *source* to the domain of *target*, can be used on the anomalous beats of *source* to generate morphologically consistent anomalous beats for *target*. To summarize, through domain adaptation, we find a morphology transformation matrix,  $\mathbf{Q}_{s \rightarrow t}$ , that adapts the abundant healthy and anomalous ECG beats of a *source* patient,  $\mathbf{s}$ , to the domain of a healthy *target* individual,  $\mathbf{t}$ , with limited healthy beats. The transformation matrix,  $\mathbf{Q}_{s \rightarrow t}$ , is strictly for transforming the beats from *source* to



**Figure 3.10** A visual view of the effect of the transformation  $Q_{s \rightarrow t}$  on  $S_s$ . As shown in Figure 3.8, most of  $S_s$  cannot be represented by  $D_t$ , thus, most of  $S_s$  lives in  $N(D_t^T)$ . After the transformation, the new beats,  $S_{s \rightarrow t}$ , can be represented by  $D_t$ , the dictionary for the healthy beats of  $t$ . Therefore, the transformation creates novel beats that are adapted from  $S_s$  to be morphologically similar to the beats of  $t$ .

---

### Algorithm 2 SR-based MTM Derivation Algorithm

---

- 1: **procedure** MTM\_DERIVATION( $D_t, S_s, \lambda, \gamma, \eta$ , iterations)
  - 2:      $Q_{s \rightarrow t} \leftarrow I_{N \times N}$
  - 3:     **for**  $it \leftarrow 1$  to iterations **do**
  - 4:          $S_{s \rightarrow t} \leftarrow Q_{s \rightarrow t} S_s$  ▷ Domain Adaptation
  - 5:          $S_{s \rightarrow t} \leftarrow S_{s \rightarrow t} \cdot \text{diag}(1 / \|(S_{s \rightarrow t})_{\cdot i}\|_2)$  ▷ Normalize Columns of  $S_{s \rightarrow t}$
  - 6:          $X_{s^*} \leftarrow \arg \min_{X_{s^*}} \|S_{s \rightarrow t} - D_t X_{s^*}\|_2^2 + \lambda \|X_{s^*}\|_1$  ▷ Solve with ADMM
  - 7:          $\nabla Q_{s \rightarrow t} \leftarrow ((1 + \gamma) Q_{s \rightarrow t} - \gamma I_{N \times N}) S_s S_s^T - D_t X_{s^*} S_s^T$  ▷ Gradient
  - 8:          $Q_{s \rightarrow t} \leftarrow Q_{s \rightarrow t} - \eta \nabla Q_{s \rightarrow t}$
  - 9:     **end for**
  - 10:     **return**  $Q_{s \rightarrow t}, X_{s^*}$
  - 11: **end procedure**
-

*target*; however, the same process can be repeated for any patient registered to the system with known healthy and anomalous beats,  $\{\mathbf{s}_i \mid i \in [1 \dots n]\}$ , to find  $n$  transformation matrices,  $\mathbf{Q}_{\mathbf{s}_i \rightarrow \mathbf{t}}$ . Therefore, we propose the following comprehensive data generation scheme to populate a large dataset for  $\mathbf{t}$ , for whom we only have a few healthy beats otherwise:

---

**Algorithm 3** Dataset Generation Algorithm using MTMs

---

```

1: procedure DATASET_GENERATION( $\mathbf{S}_t, \mathbf{S}_{\mathbf{s}_{1 \dots n}}, \mathbf{S}_{\mathbf{s}_{1 \dots n}}^A, K, \lambda, \gamma, \eta, \tau_{cvg}, \text{iters}$ )
2:    $\mathbf{D}_t \leftarrow \text{MOD\_LASSO}(\mathbf{S}_t, K, \lambda, \tau_{cvg}, \text{iters})$ 
3:   Dataset  $\leftarrow \{\mathbf{S}_t\}$ 
4:   for  $i \in 1 \dots n$  do
5:      $\mathbf{Q}_{\mathbf{s}_i \rightarrow \mathbf{t}} \leftarrow \text{MTM\_DERIVATION}(\mathbf{D}_t, \mathbf{S}_{\mathbf{s}_i}, \lambda, \gamma, \eta, \text{iters})$ 
6:      $\mathbf{S}_{\mathbf{s}_i \rightarrow \mathbf{t}} \leftarrow \text{Normalize}(\mathbf{Q}_{\mathbf{s}_i \rightarrow \mathbf{t}} \mathbf{S}_{\mathbf{s}_i})$   $\triangleright$  Domain Adaptation
7:      $\mathbf{S}_{\mathbf{s}_i \rightarrow \mathbf{t}}^A \leftarrow \text{Normalize}(\mathbf{Q}_{\mathbf{s}_i \rightarrow \mathbf{t}} \mathbf{S}_{\mathbf{s}_i}^A)$   $\triangleright$  Domain Adaptation (Anomalous)
8:     Dataset  $\leftarrow \{\text{Dataset}, \mathbf{S}_{\mathbf{s}_i \rightarrow \mathbf{t}}, \mathbf{S}_{\mathbf{s}_i \rightarrow \mathbf{t}}^A\}$ 
9:   end for
10:  return Dataset
11: end procedure

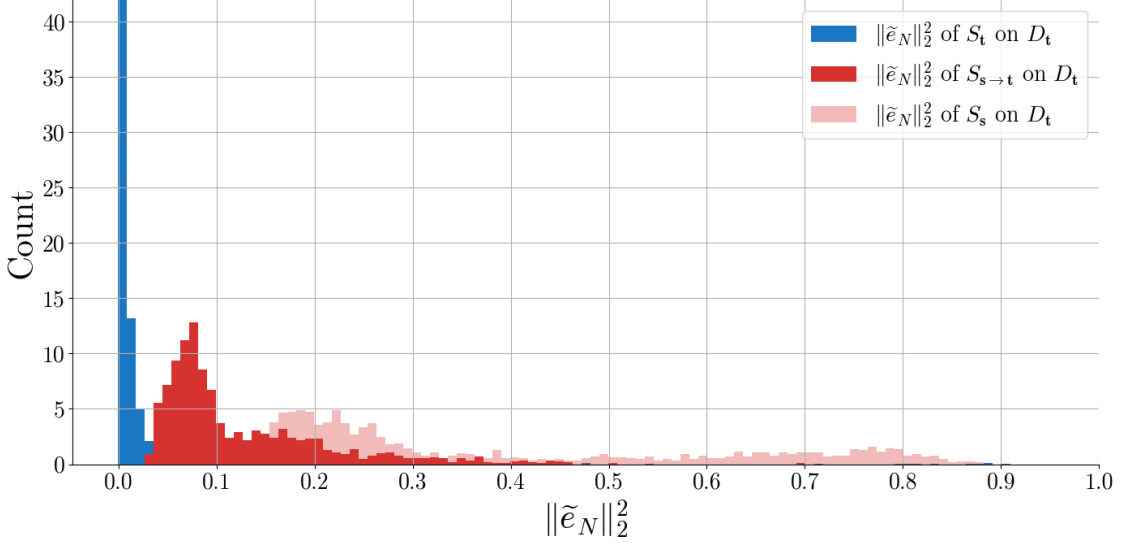
```

---

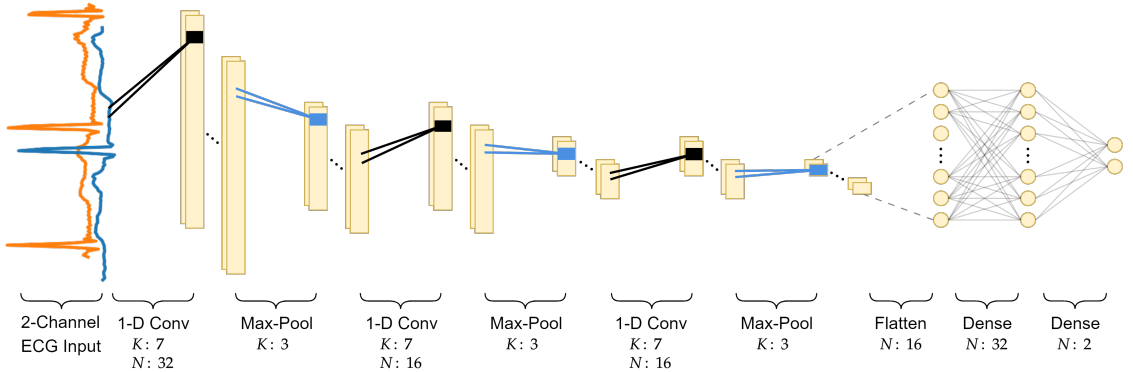
In Algorithm 3,  $\mathbf{S}_t$  and  $\mathbf{S}_{\mathbf{s}_{1 \dots n}}$  are all sets of healthy ECG beats, whereas  $\mathbf{S}_{\mathbf{s}_{1 \dots n}}^A$  are anomalous beats belonging to the source patients.

### 3.5 Personalized Domain-adapted Arrhythmia Detection

This section builds upon the findings of Section 3.4 to propose a neural network-based ECG arrhythmia detection scheme for the *target* individual after creating a morphologically consistent, large dataset using many *source* patients. First, we repeat the data generation procedure described in Algorithm 3 for beat-trios (as in Figure 2.2) to obtain two domain-adapted datasets for the *target*: a single beat dataset and a beat-trio dataset. Beat-trios introduce timescale information, which can help identify anomalies based on the location of the R-peak (or other ECG segments) in between two other R-peaks. Then, we train a convolutional neural network that takes in a single beat and its corresponding beat-trio and outputs whether this beat is anomalous. We train the same 1D CNN model as in the work [44]. The architecture of the CNN is given in Figure 3.12, and the network and training details are given in Chapter 4. The model is extremely parameter efficient with only  $6K$  parameters and can be deployed on low-end devices due to its memory and runtime efficiency. The model takes up only 24 KB of space and performs  $350K$  FLOPs per input.



**Figure 3.11** Histogram of NPE energies,  $\|\tilde{e}_N\|_2^2$ , after domain adaptation, for the domain-adapted healthy beats of source,  $S_{s \rightarrow t}$ , and healthy beats of target,  $S_t$ , on the dictionary of target,  $D_t$ . The error energies for the original beats,  $S_s$ , are provided in pale red for comparison. After domain adaptation, the representations of the healthy beats of  $s$  on  $D_t$  are much improved.



**Figure 3.12** The two-channel convolutional neural network used in all experiments.  $K$  is the filter size, and  $N$  is the number of neurons in the layer.

### 3.6 Ensemble Learning via Probabilistic Threshold Estimation

In Sections 3.2 and 3.3, we explain three different methods (namely NPE, LAE, and ARE) for efficiently calculating representation errors of ECG signals on a dictionary  $D$ . In [89], Carrera *et al.* show that representation errors can be used to classify healthy and anomalous beats by thresholding. In their work, the threshold must be determined empirically, and therefore, they report the area under the curve (AUC) of their method’s receiver operating characteristic (ROC). In this section, first, we propose a probabilistic threshold selection procedure for the simple thresholding-

based classifier built upon NPE (or either one of the error vectors). The threshold is selected as follows: first, we calculate the NPE of each heartbeat in the domain-adapted dataset of the *target* individual as described in Section 3.2. This dataset includes both healthy and anomalous beats. Next, we fit two probability distributions to the NPE of these healthy and anomalous beats using *maximum likelihood estimation* (MLE). Upon evaluation of the distributions of NPE for healthy and anomalous beats, we assumed an exponential distribution for the healthy beats and a Gaussian distribution for the anomalies (Figure 3.8 shows an example of how the NPEs of these beats are typically distributed). Finally, when classifying a new beat belonging to  $\mathbf{t}$ ,  $\mathbf{s}_t^{(i)}$ , we find its NPE through  $\mathbf{F}_t \mathbf{s}_t^{(i)}$  and calculate the likelihood of the error belonging to one of the two distributions. Thus, we create an automatic procedure to classify beats using a very simple and computationally efficient (as given in Table 3.1) thresholding-based classifier without needing to empirically tune a threshold. Since the classification is performed based on the representation error, we refer to this lightweight classifier as the **representation error-based classifier (RE-C)**.

RE-C is efficient but performs worse than the CNN described in Section 3.5. Instead of choosing one of the two and sacrificing either efficiency or performance, we merge both classifiers into an ensemble model. Whenever the confidence of the CNN is low, we consult the RE-C and classify the heartbeat to its decision. The confidence of the CNN is determined by the maximum of its two output neurons (see Figure 3.12). If the confidence is above the threshold,  $\mathcal{C}$ , then the CNN is used to classify the heartbeat. If the confidence is less than  $\mathcal{C}$ , we calculate the representation error of the heartbeat (e.g., NPE) and use the RE-C for classification.  $\mathcal{C}$  is chosen as the confidence that maximizes the F1-Score of the validation set that we split from the *target's* domain-adapted dataset. Despite introducing a threshold  $\mathcal{C}$  that needs to be selected, we show that in our experiments, any choice of  $\mathcal{C}$  results in a performance improvement (see Chapter 4).

## 4 Experiments

### 4.1 Experimental Setup

#### 4.1.1 MIT-BIH Arrhythmia Database

In this thesis, to provide a fair comparison with many related ECG works in literature [37, 44, 89, 99], we use the benchmark MIT-BIH Arrhythmia Database [48]. The database contains 48 two-channel ECG recordings from 47 different patients. Each recording is approximately half an hour long and digitized at 360 samples per second with 11-bit resolution over a 10 mV range. The recordings are independently annotated by two or more cardiologists, and the reference annotations for each beat ( $\approx 110,000$  annotations) are included with the database.

#### AAMI Standard

The MIT-BIH database is processed following the guidelines of the AAMI standard [106]. The heartbeats are combined into the following five heartbeat classes: N (beats occurring in the sinus node), V (ventricular ectopic beats), S (supraventricular ectopic beats), F (fusion beats), and Q (uncategorizable beats). During arrhythmia detection, heartbeats are classified as either normal (N) or anomalous (V, S, F, Q). The recordings from patients 102, 104, 107, and 217 are excluded because they contain paced beats as a result of a pacemaker. Patients 105, 114, 201, 202, 207, 209, 213, 222, 223, and 234 are excluded as their recordings show significant heartbeat variations. Thus, of the original 48 recordings, we use 34 records belonging to 34 different patients.

#### Preprocessing

Using the annotations provided in the MIT-BIH database, we segment the half-hour long recordings into singular heartbeats. First, we locate the R-peaks of each beat from the annotations. Then, to obtain each beat, we identify the two neighboring R-peaks and exclude 10% of the samples from each side of the interval, resulting in a segment that spans 80% of the distance between the adjacent R-peaks. Moreover, we construct beat-trios to capture the temporal and morphological characteristics of the beats. A beat-trio is obtained similar to a single beat; we identify the two neighboring R-peaks and *extend* 10% outward, resulting in a segment that contains 3 R-peaks in total. We resample each beat segment to 128 samples, remove the baseline wander, and normalize them to have unit signal energy, i.e.,  $\|\mathbf{s}\|_2^2 = 1$ .

## 4.1.2 Patient-specific Domain-adapted Datasets

### Healthy Dictionary Beat Set

After preprocessing, we have 34 recordings from 34 patients with different numbers of healthy and anomalous heartbeats, where all recordings are segmented into single beats and corresponding beat-trios. Following the AAMI recommendation, we form a small set of healthy beats for each patient  $p$ ,  $\mathbf{H}_p \in \mathbb{R}^{L \times B_p}$ , by extracting all the healthy beats from the first five minutes of their recording. The remaining twenty-five minutes of the recording and all the anomalous beats within the first five minutes (if any) form the test set for the patient  $p$ . Following the methodologies described in Section 3.1, we learn dictionaries for each patient, denoted as  $\mathbf{D}_p$ , using  $\mathbf{H}_p$ .

### Learning MTMs and Generating Datasets for each Patient

In Section 3.4, we propose the morphology transformation matrix (MTM) from a *source* patient with abundant healthy and anomalous beats to a *target* individual, for whom we only have a small set of healthy beats and no anomalous beats. In Algorithm 3, we propose a dataset generation procedure to create a large dataset with healthy and anomalous beats for the *target* individual through MTMs. In our experiments, we repeatedly apply Algorithm 3 to generate datasets for each of the 34 patients in the MIT-BIH database. At each time, a new patient  $p$  becomes the *target* (with only healthy beats,  $\mathbf{H}_p$ ), and the remaining 33 patients become the *sources* (with healthy and anomalous beats). Using the learned MTMs, we transform all healthy and anomalous beats in the half-hour recordings of the *sources* to the domain of *target* and generate *target's* dataset. As a result, we obtain  $34 \times 33 = 1122$  transformation matrices,  $\mathbf{Q}_{s_i \rightarrow t_j}$ , from each patient to every other patient, and **34 domain-adapted datasets**. We leave out 20% of the beats in each dataset for validation and the remaining 80% for training. To reiterate, the test set for each patient is their remaining twenty-five-minute recordings, which are left unseen. These patient-specific, domain-adapted datasets are used in the following experiments to train our classifiers, RE-C and a CNN.

Additionally, we generate 34 **baseline datasets** alongside the domain-adapted datasets for a fair comparison. The baseline dataset for a patient  $p$  contains their small set of healthy beats,  $\mathbf{H}_p$ , and the half-hour recordings of the remaining patients *without* domain adaptation. The baseline datasets allow us to directly see the effect of domain adaptation on arrhythmia detection performance. Once again, we separate 20% of the beats in each dataset for validation.

### 4.1.3 Network Training Setup

We train a separate CNN for each patient using their corresponding domain-adapted dataset to obtain 34 personalized CNNs in total. As given in Figure 3.12, we train a 1-D CNN with 3 convolutional layers and 2 dense layers. The convolution kernel size is set to 7, with no padding and unit stride. Each convolutional layer is followed by a max-pooling layer with a stride of 3 and a hyperbolic tangent activation. The first dense layer has a ReLU activation, and the final layer uses the softmax activation. We use 32, 16, 16, and 32 neurons in the layers, respectively, and 2 neurons in the output layer, one for each class. The model takes a two-channel ECG input: a single beat and its corresponding beat-trio. This is the same CNN architecture as in [44] for a fair comparison. We use the cross-entropy loss function and the weight-decayed Adam optimizer [109] for training. The model stops training after 15 epochs have passed without improving the validation loss. We repeat the training process 10 times with randomly initialized weights and average the results. At the end of training, the weights that obtain the lowest validation loss are used for evaluation.

### 4.1.4 Performance Metrics

At evaluation time, we obtain cumulative confusion matrices of the following form:

**Table 4.1** *The format of the obtained confusion matrices during evaluation.*

		Ground Truth	
		Arrhythmia (A)	Healthy (H)
Predicted	A	True Positives (TP)	False Positives (FP)
	H	False Negatives (FN)	True Negatives (TN)

True positives are the correctly detected anomalous heartbeats, and true negatives are the correctly detected healthy heartbeats. Since there is a separate classifier for each patient and there are 10 training runs, we report the cumulative confusion matrix (i.e., the sum of all confusion matrices). We compare the proposed and competing methods with the following performance metrics:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad (4.1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (4.2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (4.3)$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4.4)$$

Additionally, we report a separate confusion matrix for patient 232 to show the effectiveness of domain adaptation for patients with significant beat variations.

### 4.1.5 Hyperparameters

#### Number of Dictionary Atoms

We set the number of dictionary atoms,  $K$ , to 20 to compare fairly with prior art [89]. We provide this section to discuss the effect of  $K$ . The dictionary can only utilize a  $K$ -dimensional column space to represent all the healthy beats. If  $K$  is too small, the dictionary may lose its representation power, even for a patient’s small healthy dictionary set. Therefore, the representation errors will be high for both healthy and anomalous heartbeats, which will hinder the performance of RE-C, the representation error-based classifier. Furthermore, Algorithm 2 may fail to find a transformation matrix since the target dictionary is not a good subspace for healthy beats. If  $K$  is too large, the column space of the dictionary may also be too large and may be able to represent even the anomalous beats alongside the healthy beats. Figure 3.9 provides a good geometric view for the effect of the size of  $C(\mathbf{D})$ .

#### Dictionary and MTM Learning

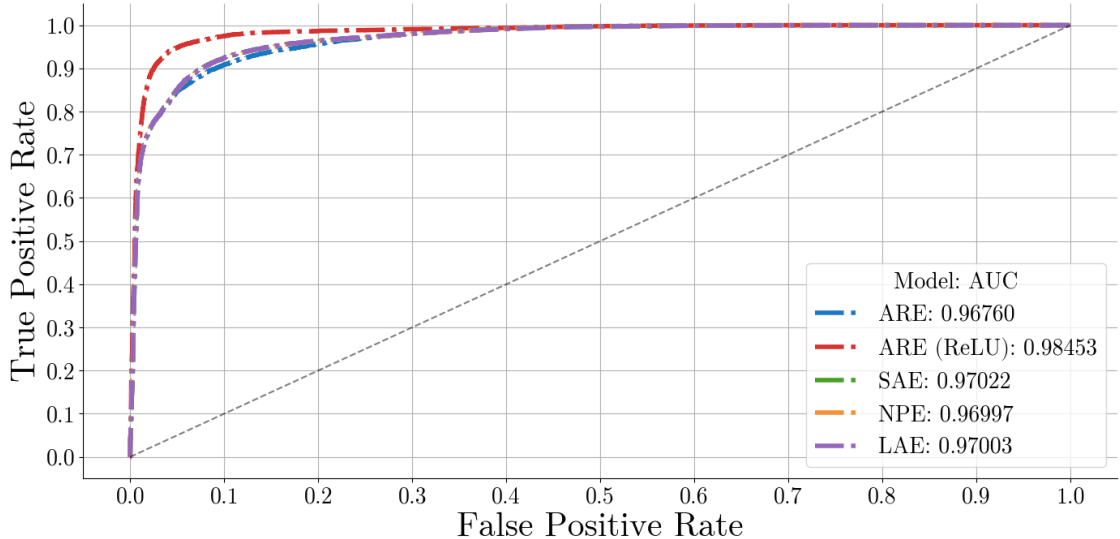
When learning the dictionaries for each patient with Algorithm 1,  $\lambda$  is set as 0.01 and  $\tau_{avg}$  as  $10^{-6}$ . When learning the MTMs with Algorithm 2,  $\gamma$  is set as 0.2 with a learning rate of  $\eta = 0.002$  and 25 iterations. The hyperparameters are empirically selected.

## 4.2 Comparisons Between Representation Errors

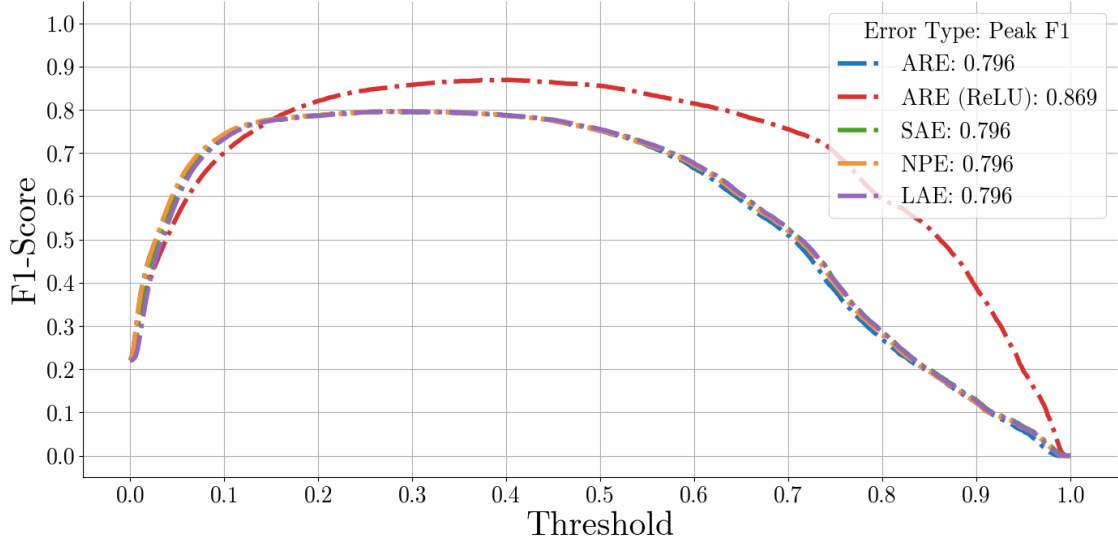
In Sections 3.2 and 3.3, we introduce three different representation errors, namely NPE, LAE, and ARE, and compare their computational complexity with the competing method [89] that calculates SAE. We show in Table 3.1 that computing the NPE is the most performant among these errors. In this section, we compare their arrhythmia detection performance by building a simple thresholding-based classifier from each error type. In their work, the authors of [89] only compute the AUC score of SAE; otherwise, an empirical threshold must be selected. We provide a way to automatically select this threshold and evaluate its performance later in Section 4.4. Now, we compare the AUC scores of NPE, LAE, and ARE with SAE. To get the AUC score, first, we calculate the representation errors of the test sets of each patient,  $\mathbf{S}_p$ , on their corresponding dictionary,  $\mathbf{D}_p$ , as follows:

$$\begin{aligned}
(SAE) \quad & \tilde{\mathbf{e}}_{\ell^1} = \mathbf{S}_p - \mathbf{D}_p \mathbf{X}_p \\
(LAE) \quad & \tilde{\mathbf{e}}_{\ell^2} = \mathbf{S}_p - \mathbf{D}_p \mathbf{L}_p \mathbf{S}_p \\
(NPE) \quad & \tilde{\mathbf{e}}_N = \mathbf{F}_p \mathbf{S}_p \\
(ARE) \quad & \tilde{\mathbf{e}}_{AE} = \mathbf{S}_p - \boldsymbol{\theta}_{\mathcal{D}_p} \phi(\boldsymbol{\theta}_{\mathcal{D}_p}^T \mathbf{S}_p).
\end{aligned} \tag{4.5}$$

Once again,  $\mathbf{F}_p$ ,  $\mathbf{L}_p$ , and  $\boldsymbol{\theta}_{\mathcal{D}_p}$  are learned beforehand according to Section 3.2 and  $\mathbf{X}_p$  is calculated using ADMM. Since the ECG beats are normalized at each step, the energy of the error vectors is at most 1, e.g.,  $\|\tilde{\mathbf{e}}_N\|_2^2 \leq 1$ . In Figure 4.1, we plot the ROC curves and report the AUC score of each error type. Our experiments show that NPE, LAE, and SAE perform equally at  $\approx 0.97$ , with ARE following just behind. Moreover, when non-linearity is introduced in the autoencoder with  $\phi = ReLU$ , the AUC score improves to  $\approx 0.985$ . As shown in Figure 3.7, *ReLU* allows the dictionary to learn a more compact representation of the ECG, focusing locally around the QRS complex. The compactness of the dictionary allows healthy and anomalous beats to be better separated with a simple threshold compared to the other error types. Afterward, in Figure 4.2, we plot the F1-Scores against all possible thresholds. Similarly, ARE with *ReLU* performs the best at  $\approx 0.87$  peak F1-Score and an overall larger area under the F1 curve. The larger area under the curve implies that many choices of a threshold will yield a robust classifier. The rest of the error types perform equally in F1-Score, with NPE being the least computationally demanding (see Table 3.1).



**Figure 4.1** ROC curves of each error type, namely SAE, LAE, NPE, ARE, and ARE (ReLU). The plots indicate that autoencoder representation error with a ReLU activation function separates the healthy and anomalous beats the best. The rest of the error types perform almost equally, with NPE being the fastest to compute among all.



**Figure 4.2** F1-Score vs. threshold of each error type, namely SAE, LAE, NPE, ARE, and ARE (ReLU). The plots show that autoencoder representation error with a ReLU activation function performs the best at  $\approx 0.87$  peak F1-Score. The rest of the error types perform equally, with NPE being the least computationally demanding.

### 4.3 Personalized and Zero-shot Arrhythmia Detection through Domain Adaptation

We generate two datasets for each patient in the MIT-BIH database, namely **baseline datasets** and **domain-adapted datasets**. We explain how these datasets are generated in Section 4.1.2. To reiterate, the baseline dataset contains the dictionary beat set of patient  $p$ ,  $\mathbf{H}_p$ , and all of the healthy and anomalous beats from the remaining patients *without* domain adaptation. On the other hand, the domain-adapted datasets contain the dictionary beat set and the healthy and anomalous beats from the remaining patients *with* domain adaptation using our MTMs. We train a separate CNN for each patient and repeat the process 10 times. After each training run, we evaluate every CNN using the corresponding patient’s test set,  $\mathbf{S}_p$ . The CNN has two output neurons, one for healthy and the other for anomalous beats. Each beat is classified according to the class of the neuron with the higher activation. Since there are 34 patients in the database and 10 independent training runs, we calculate our performance metrics from a cumulative confusion matrix. We report the confusion matrices for the baseline and domain-adapted datasets in Table 4.2.

The changes in the two confusion matrices are a direct result of domain adaptation since the two training setups are the same otherwise. We can see significant improvements due to domain adaptation. The number of correctly identified arrhythmias increases from 64447 to 72212, meaning that DA helps detect around

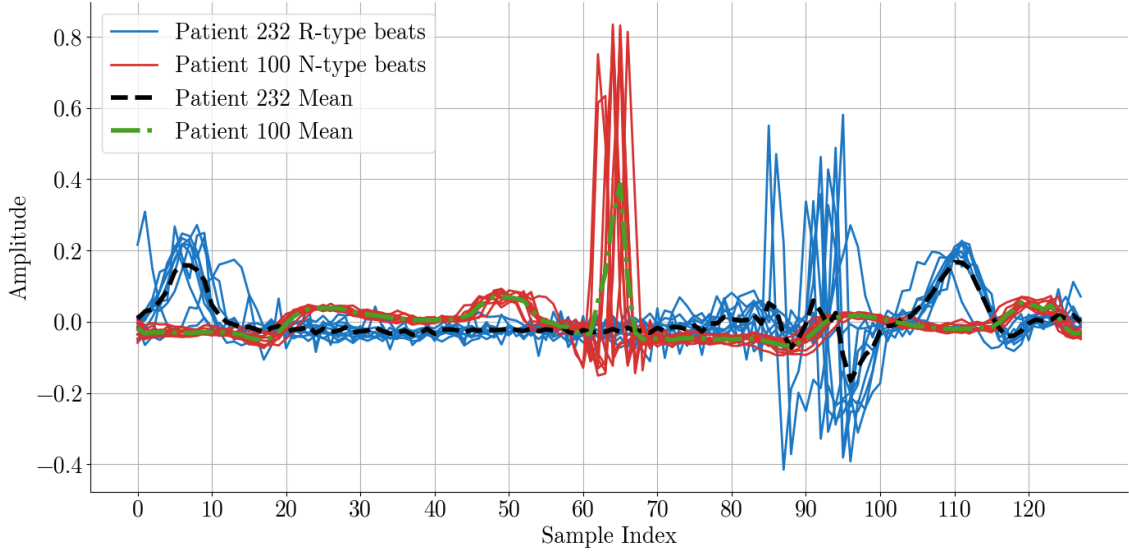
**Table 4.2** Cumulative confusion matrices for the baseline (left) and domain-adapted (right) datasets over 10 independent training runs.  $A$  = Arrhythmia,  $H$  = Healthy.

		Ground Truth					
		A		H		A	
Predicted	A	64447	7210	A	72212	7065	
	H	15183	557070	H	7418	557215	

8000 more arrhythmias accurately. As a result, we report fewer false negatives (i.e., arrhythmias that the CNN misses). On top of that, the baseline dataset achieves 96.5% accuracy and 85.2% F1-Score, while the domain-adapted dataset achieves 97.8% accuracy and 90.9% F1-Score, summarily improving the performance. The overall performance of both classifiers are reported in Table 4.4, along with our other methods. The results are compared with the competing methods later in this chapter.

In addition to reporting the accumulated results, we analyze patient-specific confusion matrices. The results indicate that domain adaptation significantly improves arrhythmia detection for patients with healthy heartbeats dissimilar to other patients' healthy heartbeats. This is due to the different ECG beat types in healthy heartbeats. The AAMI standard groups all healthy beats under the single type N, but there are different sub-types, such as the left bundle branch block beats (L), right bundle branch block beats (R), atrial escape beats (e), and nodal (junctional) escape beats (j). These beat types may show variations between them, resulting in very distinct healthy beats for one patient compared to other patients. We provide patient 232 from the MIT-BIH database as an example of a patient whose healthy beats are shown in Figure 4.3.

Patient 232's healthy beats are significantly different from other patients. Thus, directly using other patients' beats during arrhythmia detection for patient 232 negatively affects the detection performance. In Table 4.3, we provide the confusion matrices for only patient 232, *without* domain adaptation vs. *with* domain adaptation. The CNN trained on the baseline dataset has 10449 false negatives, meaning 10449 anomalous beats are being classified as healthy, and only  $\approx 25\%$  of all arrhythmias are detected by the classifier. The reason for the high number of false negatives is that healthy ECGs from other patients resemble anomalous beats for patient 232, as in Figure 4.3. The CNN is trained to classify such beats as healthy; thus, when it processes the real anomalous beats for patient 232, it detects them as healthy. Through domain adaptation, we adapt the morphology of these beats to be consistent with patient 232. Consequently, we drastically reduce the number of



**Figure 4.3** A set of N-type healthy beats from patient 100 vs. R-type healthy beats from patient 232. Despite being healthy, R-type beats have distinct morphological differences from N-type beats.

false positives from 10449 to 4756, and catch  $\approx 65\%$  of all arrhythmias instead of  $\approx 25\%$ . Arrhythmia detection for patient 232 utterly fails with a global classifier. On the other hand, domain adaptation significantly improves detection performance, demonstrating that personalized adaptation is crucial for accurate monitoring. This case study highlights the importance of tailoring classifiers to individual patients, especially in cases where inter-patient morphological variability is high.

**Table 4.3** Accumulated confusion matrices belonging to patient 232 for the baseline (left) and domain-adapted (right) datasets over 10 independent training runs. A = Arrhythmia, H = Healthy.

		Ground Truth				
		A		H		
Predicted	A	3361	185	A	9054	218
	H	10449	2965	H	4756	2932

#### 4.4 Ensemble Arrhythmia Detection

At this stage, we have access to trained CNN classifiers and the domain-adapted datasets for each patient. Now, we merge the CNN classifiers with the representation error-based classifier, RE-C. First, we estimate the parameters of the two probability distributions in the RE-C: an exponential distribution for the healthy beats and a

Gaussian distribution for the anomalous beats. The probability density functions (PDF) of these distributions are:

$$f_{exp}(x; \beta) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad f_{gauss}(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \quad (4.6)$$

The maximum likelihood estimate for the parameter  $\beta$  is:

$$\hat{\beta} = \arg \max_{\beta} \prod_{i=1}^n \frac{1}{\beta} e^{-x_i/\beta} = \frac{\sum_{i=1}^n x_i}{n}, \quad (4.7)$$

and for the parameters  $\mu$  and  $\sigma$  are:

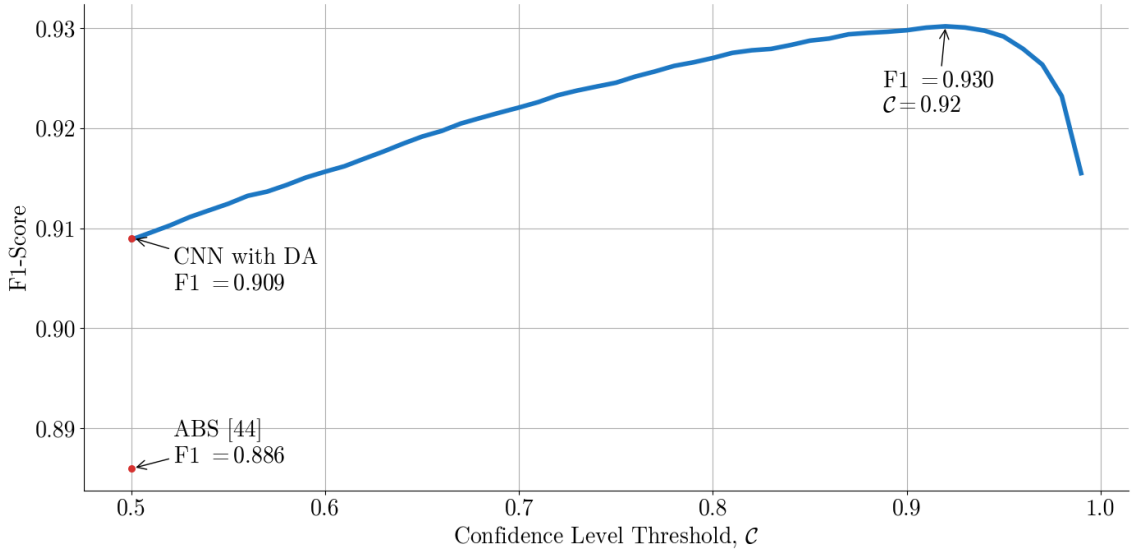
$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}. \quad (4.8)$$

Afterward, when a new ECG beat is measured, we can calculate its NPE and measure the likelihood of it belonging to either  $f_{exp}$  or  $f_{gauss}$ . The beats belonging to  $f_{exp}$  are classified as healthy, and those belonging to  $f_{gauss}$  are classified as arrhythmia. The standalone performance of the RE-C is given in Table 4.4 for the error types SAE and NPE.

Next, we evaluate the performance of the ensemble classifier. The confidence threshold,  $\mathcal{C}$ , is determined for each patient as the one that maximizes F1-Score in the validation set. If  $\mathcal{C} \leq 0.5$ , we only use the CNN as the classifier and never consult the RE-C. Likewise, if  $\mathcal{C} \geq 1$ , we only use the RE-C and never consult the CNN. Any value in between is a possible confidence threshold that offers a trade-off between the CNN and the RE-C. Nonetheless, we show in Figure 4.4 that any choice of  $\mathcal{C}$  in a broad range improves the classifier’s performance.

## 4.5 Comparisons with Competing Methods

We provide a comprehensive analysis of the proposed methods with the prior art in Table 4.4. There are few algorithms in the literature regarding personalized and zero-shot ECG arrhythmia detection. The algorithms included in the table are the abnormal beat synthesis (ABS) method proposed in [44], sparse approximation error-based detection described in [89], and GAN-based arrhythmia generation and detection given in [105]. We mark all personalized methods with a diamond ( $\diamond$ ) and all zero-shot methods with a star (\*). In addition to personalized and zero-shot methods, we include state-of-the-art CNN-based ECG classifiers [37, 100, 99], and a GAN-based ECG generation algorithm from [104].



**Figure 4.4** The F1-Score of the ensemble classifier for any choice of  $C$  compared against the state-of-the-art arrhythmia synthesis method in [44] and the standalone CNN with domain adaptation. Any choice of  $C$  in a broad range improves the performance over the CNN.

Among the CNN-based methods, [37] and [100] are personalized classifiers, and [99] is a global classifier (i.e., a single classifier for all patients). These methods rely on the set of anomalous beats from the *target* patient; therefore, they cannot be directly compared with the proposed zero-shot methods. Likewise, the GAN-based ECG generation proposed in [104] also uses anomalous beats to generate more arrhythmias. We include them in Table 4.4 to demonstrate the strength of domain adaptation, which allows our proposed zero-shot algorithms to perform better than one-shot methods. Despite using anomalous beats during training, [37, 100, 99] achieve a maximum of 89.9% F1-Score, and Shaker *et al.* achieve 92.4% on their two-stage, and 92.9% on their end-to-end classifier. On the other hand, our CNN trained on the proposed domain-adapted datasets achieves 90.9% F1-Score, and the ensemble classifier built from RE-C and the CNN achieves 92.8% F1-Score, surpassing nearly all of the aforementioned one-shot methods without using any arrhythmia from the *target* patient.

The baseline dataset, which includes all healthy and anomalous beats from the source patients *without* domain adaptation, results in 85.2% F1-Score, performing significantly behind domain adaptation with 5% F1-Score difference. This is due to variations in heartbeat morphology across patients, which domain adaptation aims to address. Moreover, Table 4.4 clearly indicates that ensemble classification with RE-C greatly improves our domain adaptation method and achieves the highest accuracy, recall, and F1-Score among all methods, improving recall by over 11% and F1-Score by over 4% compared to ABS [44].

Method	Accuracy	Specificity	Precision	Recall	F1-Score
<b>CNN</b>					
Kiranyaz <i>et al.</i> [37] $\diamond$	0.959	0.971	0.842	0.888	0.864
Zhai <i>et al.</i> [100] $\diamond$	0.968	0.976	0.879	0.920	0.899
Li <i>et al.</i> [99]	0.920	0.918	0.628	0.933	0.751
<b>GAN</b>					
Zhou <i>et al.</i> [105] $\diamond$ *	0.979	0.989	0.908	0.897	0.902
Shaker <i>et al.</i> Two-stage [104]	0.986	0.988	0.886	0.964	0.924
Shaker <i>et al.</i> End-to-end [104]	0.987	0.990	0.901	0.959	0.929
<b>SR-based (RE-C) <math>\diamond</math>*</b>					
SAE-based	0.947	0.968	0.779	0.794	0.786
NPE-based (ours)	0.947	0.968	0.779	0.794	0.786
<b>CNN <math>\diamond</math>*</b>					
ABS [44]	0.977	<b>0.995</b>	<b>0.956</b>	0.825	0.886
Baseline (ours)	0.965	0.987	0.899	0.809	0.852
Domain Adaptation (ours)	0.978	0.987	0.911	0.907	0.909
Ensemble (ours)	<b>0.982</b>	0.988	0.919	<b>0.937</b>	<b>0.928</b>
Ensemble (avg.) (ours)	0.981	0.988	0.918	0.926	0.922
Energy-efficient (40%) (ours)	0.973	0.990	0.920	0.859	0.888

**Table 4.4** Comparison of the proposed methods, namely RE-C with NPE, Baseline, Domain Adaptation, Ensemble Classification, and Energy-efficient Classification, with the competing methods. The detection performances of prior art, including global and one-shot classifiers, are presented. The results show that our personalized zero-shot ensemble model surpasses all the other methods in F1-Score and is on par with [104], even though [104] is a global one-shot GAN-based classifier with a signal length of 300 (as opposed to 128). The ensemble classifier’s confidence threshold,  $\mathcal{C}$ , is chosen using the validation set. The average ensemble classifier is the average performance over all possible confidence thresholds.

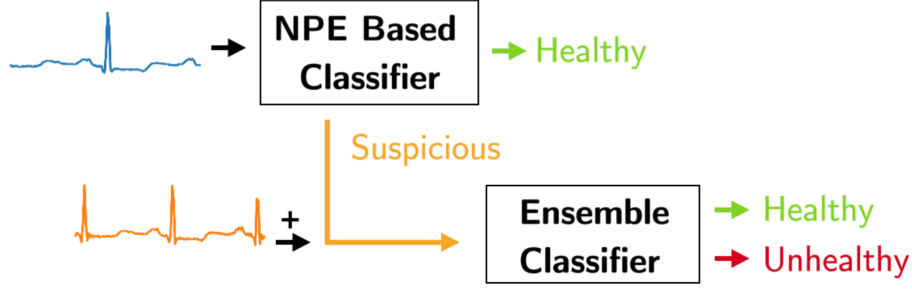
$\diamond$  Personalized classifiers.

\* Zero-shot classifiers.

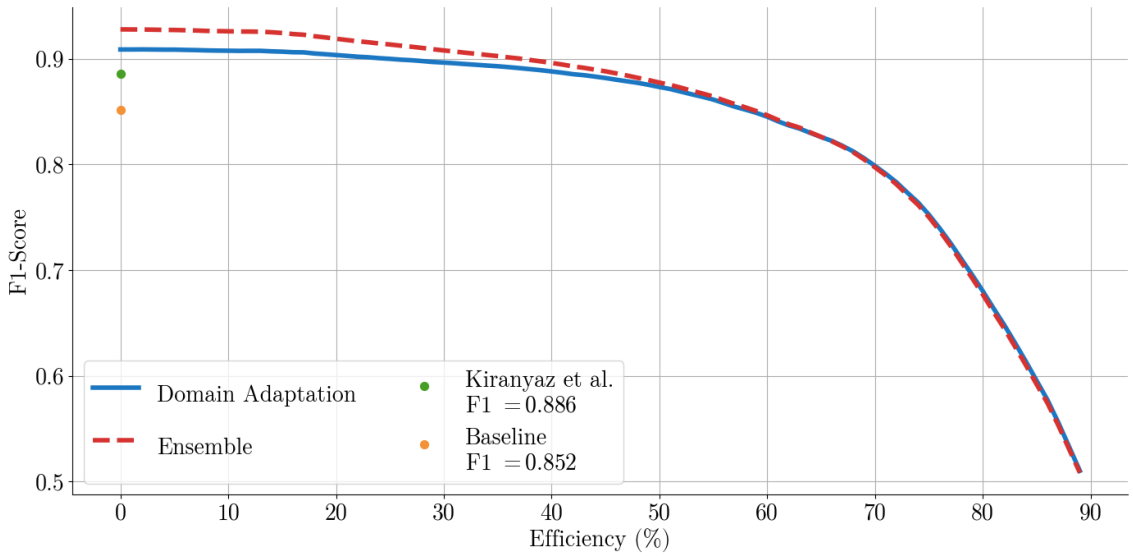
## 4.6 Energy Efficient Monitoring through Sparse Representation

We dedicate this section to describing and evaluating an energy-efficient and continuous ECG monitoring system to be deployed on mobile wear with the proposed methods described in Chapter 3. Cardiac monitoring on wearable devices such as a smartwatch has the potential to improve personalized care and provide early diagnosis for cardiovascular diseases. The current state of robust neural network-based solutions for arrhythmia detection has the downside of being energy inefficient and unsuitable for mobile wear [89, 110].

Our proposed system is built on the observation that healthy ECG beats can be represented with very little representation error on the person-specific dictionary



**Figure 4.5** Energy efficiency over F1-Scores, where the x-axis shows the percentage of test samples that are classified solely based on NPE, hence being more computationally efficient.



**Figure 4.6** Energy efficiency over F1-Scores, where the x-axis shows the percentage of test samples that are classified solely based on NPE, hence being more computationally efficient.

(see Figure 3.8). Therefore, a large number of heartbeats can be immediately classified as healthy with high confidence based on their representation error. Moreover, a continuous monitoring device is likely to encounter healthy beats through most of its monitoring lifetime and occasionally needs to detect anomalies. To this end, we propose a practical ECG monitoring system depicted in Figure 4.5. The system utilizes our representation error-based classifier (RE-C) built on the nullspace projection error (NPE) since NPE is highly energy efficient compared to other methods (see Table 3.1). Any newly measured heartbeat first passes through RE-C to detect whether it is healthy or suspicious. Then, if the beat is suspicious, it is forwarded to the proposed ensemble classifier with its corresponding beat-trio for a more detailed analysis. This simple system allows up to 40% of all the 64391 heartbeats in the 34 patients' test datasets to be immediately classified as healthy without significantly

affecting the F1-Score. We show the overall performance of the system in Figure 4.6. The horizontal axis shows the percentage of the beats classified solely using RE-C (i.e., efficiency). We show that through our efficient monitoring system, a more heavyweight classifier (e.g., the ensemble classifier) needs to be used only around 50% of the time, while the rest of the time, RE-C performs the monitoring.

## 5 Conclusion

In this thesis, we address the problem of personalized and zero-shot ECG arrhythmia detection. We provide a complete pipeline of continuous ECG monitoring for the following realistic scenario: A healthy individual registers to the monitoring system, and only a small set of healthy beats are acquired for continuous monitoring. The pipeline works by learning person-specific dictionaries and morphology transformation matrices and adapting the heartbeats of existing patients in the system to the domain of the newly registered individual. We approach the problem from the perspective of sparse representation and propose novel ideas through sparse coding, autoencoder-based dictionary learning, sparse representation-based domain adaptation, and an energy-efficient ensemble classification and arrhythmia monitoring system. Our proposed methods achieve state-of-the-art performance on the benchmark MIT-BIH Arrhythmia Database. Through rigorous experimentation on the proposed methods, we conclude this thesis with the following remarks.

### Efficient Representation Error-based Arrhythmia Detection

Given a patient  $p$  and their dictionary  $\mathbf{D}_p$ , we can construct its left annihilator matrix,  $\mathbf{F}_p$ . When we obtain a new ECG for this patient, called  $\mathbf{s}_p$ , instead of solving for the sparse coefficient vector through computationally expensive and iterative methods, such as OMP or ADMM, we use  $\mathbf{F}_p$  to directly project the error component of  $\mathbf{s}_p$  onto the left nullspace of  $\mathbf{D}_p$ . In Chapter 4, we demonstrate that **nullspace projection error**-based classification provides nearly 20 times faster computations without any performance drawbacks. Moreover, we provide an alternative method to calculate the **least-squares approximation error** that performs equally well with similar computational costs. The choice between NPE and LAE depends on the computational environment and hardware configurations. We provide the computational analysis in Table 3.1 and their respective performances in Figures 4.1 and 4.2. We conclude that NPE and LAE can be used for efficient arrhythmia detection with only a few FLOPs through simple thresholding on the representation error energy.

### Compact Dictionary Learning via Personalized Autoencoders

In Section 3.3, we show that we can train a symmetrical single hidden-layer autoencoder With only a small set of healthy heartbeats from an individual. The weights of this autoencoder correspond to the personalized dictionary for this individual.

Autoencoder-based dictionaries provide many benefits. First, they allow the representation errors to be efficiently calculated at the encoding stage with a few FLOPs (see Table 3.1). Second, we can build non-linear dictionaries through the use of non-linear functions on the activations. The non-linearity makes the dictionary learn a more compact representation of the ECG and focus locally around the QRS complex (see Figure 3.7). In return, the compactness allows healthy and anomalous beats to be better separated with a simple threshold compared to the other error types, as shown in Figures 4.1 and 4.2.

### **Sparse Representation-based Domain Adaptation through MTMs**

In Section 3.4, we propose morphology transformation matrices between *source* and *target* patients to adapt the ECG morphology of the *source* patient to *target*. Using MTMs, we can perform inter-patient beat transfer. Thus, we create large domain-adapted training datasets for each user by transferring the healthy and abnormal beats from all of the remaining users. Our CNN architecture trained on the baseline dataset only achieves 85.2% F1-Score, while the same network trained on the domain-adapted dataset achieves 90.9% F1-Score. We eliminate the need to generate synthetic abnormal beats through domain adaptation and drastically improve personalized and zero-shot arrhythmia detection performance.

### **Ensemble Arrhythmia Detection**

By fusing our RE-C and CNN, we create an ensemble classifier. Depending on the confidence of the CNN, we consult the RE-C and detect arrhythmias with both models. In Figure 4.4, we show that the ensemble classifier improves the detection performance for any choice of the confidence threshold in a broad range. Nonetheless, we choose the confidence threshold from the validation set and achieve state-of-the-art detection performance with 98.2% accuracy and 92.8% F1-Score. The results are presented in Table 4.4.

### **Energy Efficient Arrhythmia Monitoring**

Lastly, we propose an energy-efficient monitoring system that utilizes the computational efficiency of NPE and the superior performance of the ensemble classifier. The idea behind such a system is that many healthy beats will be represented in the dictionary with a small representation error and can be classified through NPE. The system outline is provided in Figure 4.5. The proposed monitoring system can classify up to 40% of all test samples solely based on the NPE without sacrificing noticeably from performance. Therefore, we conclude that with sparse representation-based domain adaptation and ensemble classification, energy-efficient monitoring for

mobile devices achieves state-of-the-art results for ECG arrhythmia detection.

## References

- [1] P. Kligfield *et al.*, “Recommendations for the standardization and interpretation of the electrocardiogram: Part i: The electrocardiogram and its technology: A scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology,” *Circulation*, vol. 115, no. 10, pp. 1306–1324, 2007.
- [2] W. B. Fye, “A history of the origin, evolution, and impact of electrocardiography,” *The American journal of cardiology*, vol. 73, no. 13, pp. 937–949, 1994.
- [3] Y. Sattar and L. Chhabra, “Electrocardiogram,” in *StatPearls [Internet]*, StatPearls Publishing, 2023.
- [4] A. Galli, F. Ambrosini, and F. Lombardi, “Holter monitoring and loop recorders: From research to clinical practice,” *Arrhythmia & electrophysiology review*, vol. 5, no. 2, p. 136, 2016.
- [5] N. Isakadze and S. S. Martin, “How useful is the smartwatch ecg?” *Trends in cardiovascular medicine*, vol. 30, no. 7, pp. 442–448, 2020.
- [6] G. Tse, “Mechanisms of cardiac arrhythmias,” *Journal of arrhythmia*, vol. 32, no. 2, pp. 75–81, 2016.
- [7] N. T. Srinivasan and R. J. Schilling, “Sudden cardiac death and arrhythmias,” *Arrhythmia & electrophysiology review*, vol. 7, no. 2, p. 111, 2018.
- [8] A. S. Adabag, R. V. Luepker, V. L. Roger, and B. J. Gersh, “Sudden cardiac death: Epidemiology and risk factors,” *Nature Reviews Cardiology*, vol. 7, no. 4, pp. 216–225, 2010.
- [9] J. Poorolajal, “Neglected major causes of death much deadlier than covid-19,” *Journal of Research in Health Sciences*, vol. 20, no. 2, e00478, 2020.
- [10] S. Kaptoge *et al.*, “World health organization cardiovascular disease risk charts: Revised models to estimate risk in 21 global regions,” *The Lancet global health*, vol. 7, no. 10, e1332–e1345, 2019.
- [11] S. S. Martin *et al.*, “2024 heart disease and stroke statistics: A report of us and global data from the american heart association,” *Circulation*, vol. 149, no. 8, e347–e913, 2024.
- [12] M. Merone, P. Soda, M. Sansone, and C. Sansone, “Ecg databases for biometric systems: A systematic review,” *Expert Systems with Applications*, vol. 67, pp. 189–202, 2017.

- [13] R. Hoekema, G. Uijen, and A. van Oosterom, “Geometrical aspects of the interindividual variability of multilead ecg recordings,” *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 5, pp. 551–559, 2001. DOI: 10.1109/10.918594.
- [14] R. D. Labati, E. Muñoz, V. Piuri, R. Sassi, and F. Scotti, “Deep-ecg: Convolutional neural networks for ecg biometric recognition,” *Pattern Recognition Letters*, vol. 126, pp. 78–85, 2019.
- [15] N. Ibtehaz *et al.*, “Edith: Ecg biometrics aided by deep learning for reliable individual authentication,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 4, pp. 928–940, 2021.
- [16] B.-H. Kim and J.-Y. Pyun, “Ecg identification for personal authentication using lstm-based deep recurrent neural networks,” *Sensors*, vol. 20, no. 11, p. 3069, 2020.
- [17] Y. N. Singh and P. Gupta, “Ecg to individual identification,” in *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, IEEE, 2008, pp. 1–8.
- [18] J. Behar, J. Oster, Q. Li, and G. D. Clifford, “Ecg signal quality during arrhythmia and its application to false alarm reduction,” *IEEE transactions on biomedical engineering*, vol. 60, no. 6, pp. 1660–1666, 2013.
- [19] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” *IEEE transactions on biomedical engineering*, no. 3, pp. 230–236, 1985.
- [20] P. De Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ecg morphology and heartbeat interval features,” *IEEE transactions on biomedical engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.
- [21] S. Aziz, S. Ahmed, and M.-S. Alouini, “Ecg-based machine-learning algorithms for heartbeat classification,” *Scientific reports*, vol. 11, no. 1, p. 18 738, 2021.
- [22] I. Beraza and I. Romero, “Comparative study of algorithms for ecg segmentation,” *Biomedical Signal Processing and Control*, vol. 34, pp. 166–173, 2017.
- [23] M. Gabbouj *et al.*, “Robust peak detection for holter ecgs by self-organized operational neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 9363–9374, 2022.
- [24] P. De Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ecg morphology and heartbeat interval features,” *IEEE transactions on biomedical engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.

- [25] P. De Chazal and R. B. Reilly, "A patient-adapting heartbeat classifier using ecg morphology and heartbeat interval features," *IEEE transactions on biomedical engineering*, vol. 53, no. 12, pp. 2535–2543, 2006.
- [26] P. De Chazal, "Detection of supraventricular and ventricular ectopic beats using a single lead ecg," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2013, pp. 45–48.
- [27] M. Llamedo and J. P. Martínez, "Heartbeat classification using feature selection driven by database generalization criteria," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 616–625, 2010.
- [28] C. Ye, B. V. Kumar, and M. T. Coimbra, "Heartbeat classification using morphological and dynamic features of ecg signals," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2930–2941, 2012.
- [29] J. L. Willems and E. Lesaffre, "Comparison of multigroup logistic and linear discriminant ecg and vcg classification," *Journal of electrocardiology*, vol. 20, no. 2, pp. 83–92, 1987.
- [30] X. Jiang, L. Zhang, Q. Zhao, and S. Albayrak, "Ecg arrhythmias recognition system based on independent component analysis feature extraction," in *TENCON 2006-2006 IEEE Region 10 Conference*, IEEE, 2006, pp. 1–4.
- [31] L. Senhadji, G. Carrault, J. Bellanger, and G. Passariello, "Comparing wavelet transforms for recognizing cardiac patterns," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 2, pp. 167–173, 1995.
- [32] K.-i. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network," *IEEE transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 179–185, 1999.
- [33] T. Ince, S. Kiranyaz, and M. Gabbouj, "A generic and robust system for automated patient-specific classification of ecg signals," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1415–1426, 2009.
- [34] I. Christov, I. Jekova, and G. Bortolan, "Premature ventricular contraction classification by the kth nearest-neighbours rule," *Physiological measurement*, vol. 26, no. 1, p. 123, 2005.
- [35] S. Osowski, L. T. Hoai, and T. Markiewicz, "Support vector machine-based expert system for reliable heartbeat recognition," *IEEE transactions on biomedical engineering*, vol. 51, no. 4, pp. 582–589, 2004.
- [36] W. Jiang and S. G. Kong, "Block-based neural networks for personalized ecg signal classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1750–1761, 2007.

- [37] S. Kiranyaz, T. Ince, and M. Gabbouj, “Real-time patient-specific ecg classification by 1-d convolutional neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2016.
- [38] J. Malik, O. C. Devecioglu, S. Kiranyaz, T. Ince, and M. Gabbouj, “Real-time patient-specific ecg classification by 1d self-operational neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 5, pp. 1788–1801, 2021.
- [39] Y. Wang, L. Sun, and S. Subramani, “Cab: Classifying arrhythmias based on imbalanced sensor data,” *KSIIT Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 7, pp. 2304–2320, 2021.
- [40] T. Golany, D. Freedman, and K. Radinsky, “Ecg ode-gan: Learning ordinary differential equations of ecg dynamics via generative adversarial learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 134–141.
- [41] T. Golany, K. Radinsky, and D. Freedman, “Simgans: Simulator-based generative adversarial networks for ecg synthesis to improve deep ecg classification,” in *International Conference on Machine Learning*, PMLR, 2020, pp. 3597–3606.
- [42] P. Wang, B. Hou, S. Shao, and R. Yan, “Ecg arrhythmias detection using auxiliary classifier generative adversarial network and residual network,” *Ieee Access*, vol. 7, pp. 100 910–100 922, 2019.
- [43] T. Golany and K. Radinsky, “Pgans: Personalized generative adversarial networks for ecg synthesis to improve patient-specific deep ecg classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 557–564.
- [44] S. Kiranyaz, T. Ince, and M. Gabbouj, “Personalized monitoring and advance warning system for cardiac arrhythmias,” *Scientific reports*, vol. 7, no. 1, p. 9270, 2017.
- [45] A. Parsi, “Improved cardiac arrhythmia prediction based on heart rate variability analysis,” *arXiv preprint arXiv:2206.03222*, 2022.
- [46] M. Malik and A. J. Camm, “Heart rate variability,” *Clinical cardiology*, vol. 13, no. 8, pp. 570–576, 1990.
- [47] C. M. van Ravenswaaij-Arts, L. A. Kollee, J. C. Hopman, G. B. Stoeltinga, and H. P. van Geijn, “Heart rate variability,” *Annals of internal medicine*, vol. 118, no. 6, pp. 436–447, 1993.

- [48] G. B. Moody and R. G. Mark, “The impact of the mit-bih arrhythmia database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [49] M. Elad, *Sparse and Redundant Representations*. Springer, Jan. 2010. DOI: 10.1007/978-1-4419-7011-4\_16.
- [50] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [51] G. Strang, “The fundamental theorem of linear algebra,” *The American Mathematical Monthly*, vol. 100, no. 9, pp. 848–855, 1993.
- [52] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [53] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of 27th Asilomar conference on signals, systems and computers*, IEEE, 1993, pp. 40–44.
- [54] G. Kutyniok, “Theory and applications of compressed sensing,” *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 79–101, 2013.
- [55] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.
- [56] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689–696.
- [57] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.
- [58] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [59] D.-S. Pham and S. Venkatesh, “Joint learning and dictionary construction for pattern recognition,” in *2008 IEEE conference on computer vision and pattern recognition*, IEEE, 2008, pp. 1–8.
- [60] Q. Zhang and B. Li, “Discriminative k-svd for dictionary learning in face recognition,” in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 2691–2698.

- [61] M. Yang, L. Zhang, X. Feng, and D. Zhang, “Fisher discrimination dictionary learning for sparse representation,” in *2011 international conference on computer vision*, IEEE, 2011, pp. 543–550.
- [62] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 791–804, 2011.
- [63] K. Engan, S. O. Aase, and J. H. Husoy, “Method of optimal directions for frame design,” in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, IEEE, vol. 5, 1999, pp. 2443–2446.
- [64] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006. DOI: 10.1109/TSP.2006.881199.
- [65] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.
- [66] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm,” *IEEE Transactions on signal processing*, vol. 45, no. 3, pp. 600–616, 1997.
- [67] K. Kreutz-Delgado and B. D. Rao, “Focuss-based dictionary learning algorithms,” in *Wavelet Applications in Signal and Image Processing VIII*, SPIE, vol. 4119, 2000, pp. 459–473.
- [68] K. Engan, B. D. Rao, and K. Kreutz-Delgado, “Frame design using focuss with method of optimal directions (mod),” in *Proc. NOR SIG*, vol. 99, 1999, pp. 65–69.
- [69] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.
- [70] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [71] R. Salakhutdinov and G. Hinton, “Deep boltzmann machines,” in *Artificial intelligence and statistics*, PMLR, 2009, pp. 448–455.
- [72] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa, “Deep dictionary learning,” *IEEE Access*, vol. 4, pp. 10 096–10 109, 2016.

- [73] H. Bourlard and Y. Kamp, “Auto-association by multilayer perceptrons and singular value decomposition,” *Biological cybernetics*, vol. 59, no. 4, pp. 291–294, 1988.
- [74] A. Makhzani and B. Frey, “K-sparse autoencoders,” *arXiv preprint arXiv:1312.5663*, 2013.
- [75] H. Zheng, H. Yong, and L. Zhang, “Deep convolutional dictionary learning for image denoising,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 630–641.
- [76] M. Yamaç, M. Ahishali, S. Kiranyaz, and M. Gabbouj, “Convolutional sparse support estimator network (csen): From energy-efficient support estimation to learning-aided compressive sensing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 290–304, 2021.
- [77] K. Cho, “Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images,” in *International conference on machine learning*, PMLR, 2013, pp. 432–440.
- [78] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive autoencoders: Explicit invariance during feature extraction,” in *Proceedings of the 28th international conference on international conference on machine learning*, 2011, pp. 833–840.
- [79] A. Majumdar and R. Ward, “Robust greedy deep dictionary learning for eeg arrhythmia classification,” in *2017 International joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 4400–4407.
- [80] H. Tang, H. Liu, W. Xiao, and N. Sebe, “When dictionary learning meets deep learning: Deep dictionary learning and coding network for image recognition with limited data,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 5, pp. 2129–2141, 2020.
- [81] Y. Li, Y. Chai, and H. Yin, “Autoencoder embedded dictionary learning for nonlinear industrial process fault diagnosis,” *Journal of Process Control*, vol. 101, pp. 24–34, 2021.
- [82] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008.
- [83] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification via kernel sparse representation,” *IEEE Transactions on Geoscience and Remote sensing*, vol. 51, no. 1, pp. 217–231, 2012.

- [84] B. Wei, M. Yang, Y. Shen, R. Rana, C. T. Chou, and W. Hu, “Real-time classification via sparse representation in acoustic sensor networks,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013, pp. 1–14.
- [85] G. Wu *et al.*, “Sparse representation-based radiomics for the diagnosis of brain tumors,” *IEEE transactions on medical imaging*, vol. 37, no. 4, pp. 893–905, 2017.
- [86] L. Zhang, M. Yang, and X. Feng, “Sparse representation or collaborative representation: Which helps face recognition?” In *2011 International conference on computer vision*, IEEE, 2011, pp. 471–478.
- [87] H. F. Huang, G. S. Hu, and L. Zhu, “Sparse representation-based heartbeat classification using independent component analysis,” *Journal of medical systems*, vol. 36, pp. 1235–1247, 2012.
- [88] S. M. Mathews, L. F. Polania, and K. E. Barner, “Leveraging a discriminative dictionary learning algorithm for single-lead ecg classification,” in *2015 41st Annual Northeast Biomedical Engineering Conference (NEBEC)*, IEEE, 2015, pp. 1–2.
- [89] D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi, “Ecg monitoring in wearable devices by sparse models,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part III 16*, Springer, 2016, pp. 145–160.
- [90] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, “Sparse coding with anomaly detection,” *Journal of Signal Processing Systems*, vol. 79, pp. 179–188, 2015.
- [91] A. L. Goldberger *et al.*, “Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, e215–e220, 2000.
- [92] D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi, “Online anomaly detection for long-term ecg monitoring using wearable devices,” *Pattern Recognition*, vol. 88, pp. 482–492, 2019.
- [93] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, “Clustering ecg complexes using hermite functions and self-organizing maps,” *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 838–848, 2000.

- [94] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, “Convolutional neural networks for patient-specific ecg classification,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 2608–2611.
- [95] M. Chourasia, A. Thakur, S. Gupta, and A. Singh, “Ecg heartbeat classification using cnn,” in *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, 2020, pp. 1–6.
- [96] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, *et al.*, “Long short term memory networks for anomaly detection in time series.,” in *Esann*, vol. 2015, 2015, p. 89.
- [97] K. Weimann and T. O. Conrad, “Transfer learning for ecg classification,” *Scientific reports*, vol. 11, no. 1, p. 5251, 2021.
- [98] T. Jun, H. Nguyen, D. Kang, D. Kim, D. Kim, and Y. Kim, “Ecg arrhythmia classification using a 2-d convolutional neural network. arxiv 2018,” *arXiv preprint arXiv:1804.06812*, vol. 10, 2018.
- [99] F. Li, Y. Xu, Z. Chen, and Z. Liu, “Automated heartbeat classification using 3-d inputs based on convolutional neural network with multi-fields of view,” *IEEE Access*, vol. 7, pp. 76 295–76 304, 2019.
- [100] X. Zhai and C. Tin, “Automated ecg classification using dual heartbeat coupling based on convolutional neural network,” *IEEE Access*, vol. 6, pp. 27 465–27 472, 2018.
- [101] A. M. Delaney, E. Brophy, and T. E. Ward, “Synthesis of realistic ecg using generative adversarial networks,” *arXiv preprint arXiv:1909.09150*, 2019.
- [102] N. Wulan, W. Wang, P. Sun, K. Wang, Y. Xia, and H. Zhang, “Generating electrocardiogram signals by deep learning,” *Neurocomputing*, vol. 404, pp. 122–136, 2020.
- [103] F. Zhu, F. Ye, Y. Fu, Q. Liu, and B. Shen, “Electrocardiogram generation with a bidirectional lstm-cnn generative adversarial network,” *Scientific reports*, vol. 9, no. 1, p. 6734, 2019.
- [104] A. M. Shaker, M. Tantawi, H. A. Shedeed, and M. F. Tolba, “Generalization of convolutional neural networks for ecg classification using generative adversarial networks,” *IEEE Access*, vol. 8, pp. 35 592–35 605, 2020.
- [105] Z. Zhou, X. Zhai, and C. Tin, “Fully automatic electrocardiogram classification system based on generative adversarial network with auxiliary classifier,” *Expert Systems with Applications*, vol. 174, p. 114 809, 2021.

- [106] A. for the Advancement of Medical Instrumentation, “Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms,” *Arlington, VA*, 1987.
- [107] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [108] G. Strang, *Linear algebra and learning from data*. SIAM, 2019.
- [109] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [110] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, “Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.