University of Calabria

Department of Computer Science, Modeling, Electronics and Systems Engineering

PhD in Information and Communication Technologies

Ph.D. Thesis

Compressive Sensing for Secure and Energy Efficient ECG Signals in Healthcare Applications

Candidate:
Bharat Lal

BHARAT LAL
24.01.2025 18:13:50
GMT+01:00

Supervisors:
Prof. Raffaele Gravina
Prof. Pasquale Corsonello

Raffaele Gravina
24.01.2025
23:30:19
GMT+02:00

Pasquale
Corsonello
27.01.2025
10:20:11
GMT+02:00

PhD Coordinator:
Prof. Giancarlo Fortino

GIANCARLO FORTINO
27.01.2025 13:37:14
GMT+02:00

Academic Year 2024/2025 (XXXVII Cycle)

To my parents, whose unwavering love, sacrifices, and belief in me have been my greatest source of strength.
To every child from underrepresented areas who dreams of getting an education and changing the world.
To all those who supported me throughout my life.
To those who believe in hard work, kindness, empathy, and the collective well-being and welfare of all living beings in the universe.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Raffaele Gravina and Prof. Pasquale Corsonello, for their consistent support, insightful guidance, and encouragement throughout my Ph.D journey. Their expertise, constructive feedback, and mentorship have been invaluable in shaping my research and academic growth.

I am also immensely grateful to Dr. Miguel Heredia Conde, my advisor during the research exchange period at the Center for Sensor Systems (ZESS), University of Siegen, Germany. His insightful guidance and collaborative work during my stay enriched my research experience and contributed significantly to this research.

My sincere thanks extend to Prof. Giancarlo Fortino, the Ph.D. coordinator, for his valuable guidance and support, especially during the initial stages of my Ph.D. In addition, I would like to thank Mehran UET Jamshoro, Pakistan for granting me study leave to complete the Ph.D. degree.

I would like to extend my heartfelt appreciation to the University of Calabria, for their financial support and the scholarship provided throughout my Ph.D.

I am deeply grateful to my family for their unconditional love, patience, and support. Specially to my elder brother Dr. Shankar lal Meghwar and younger brothers Engr. Munesh Meghwar and Dr. Hakam Das for their support in most difficult times during Ph.D.

To my friends, colleagues and collaborators, I am truly thankful for your valuable support in difficult time, shared insights, and camaraderie. The discussions, brainstorming sessions, and encouragement you provided have greatly contributed to the completion of my thesis.

Abstract

In recent years, there has been a significant focus on advancing continuous health monitoring solutions through wireless technologies, particularly wearable devices, which facilitate the remote analysis of physiological data. While these innovations have revolutionized healthcare, they also present substantial challenges, including the transmission of large data volumes within the constraints of devices with limited battery life. Addressing these challenges requires the development of energy-efficient, secure, and diagnostically accurate telemonitoring systems, incorporating strategies to reduce data transmission while supporting real-time analysis.

This thesis explores the integration of compressive sensing and compressed learning frameworks to address these challenges, with a specific focus on electrocardiogram (ECG) signal processing. By leveraging the inherent sparsity of biomedical signals, this research develops innovative methodologies to achieve secure data transmission, efficient signal reconstruction, and accurate diagnosis. The proposed frameworks balance theoretical advancements with practical implementations, bridging gaps in wearable healthcare technology and remote monitoring systems.

Key contributions include the design of a dual-layer cryptographic framework for secure and energy-efficient ECG data transmission, validated on low-power hardware platforms. The thesis also introduces a data-driven sensing matrix and overcomplete dictionaries tailored to ECG signals, enabling high compression ratios without compromising reconstruction quality. Further, it integrates deep learning into the CS framework, proposing autoencoder-based models for efficient sensing and reconstruction. These models dynamically adapt to data, achieving superior performance in accuracy and computational efficiency with respect to the state-of-the-art methods.

Extending beyond traditional reconstruction, the research develops a compressed learning framework that directly classifies ECG signals in the compressed domain, bypassing the reconstruction step. This approach significantly reduces computational overhead while maintaining diagnostic accuracy, with hardware validation demonstrating its feasibility for real-world applications.

The findings highlight the transformative potential of compressive sensing and compressed learning in enabling scalable, secure, and energy-efficient healthcare solutions. This thesis sets a robust foundation for next-generation wearable technologies and remote monitoring systems, contributing to improved healthcare accessibility and diagnostic reliability. Future research directions include extending these methodologies to other physiological signals, optimizing hardware implementations, and exploring multimodal health monitoring systems.

Sommario

Negli ultimi anni, si è posta una significativa attenzione sullo sviluppo di soluzioni avanzate per il monitoraggio continuativo della salute attraverso tecnologie wireless, in particolare dispositivi indossabili, che facilitano l'analisi remota dei segnali fisiologici. Sebbene queste innovazioni abbiano rivoluzionato il settore sanitario, presentano anche sfide rilevanti, tra cui l'elaborazione in tempo reale e la trasmissione di grandi volumi di dati su dispositivi con risorse limitate, tra cui, non per ultimo, capacità fortemente limitata della batteria. Affrontare queste sfide richiede lo sviluppo di sistemi di telemonitoraggio energeticamente efficienti, sicuri e clinicamente accurati, che adottino strategie per ridurre la trasmissione dei dati supportando al contempo l'analisi in tempo reale.

Questa tesi esplora l'integrazione dei framework compressive sensing e compressed learning per affrontare queste sfide, con un'attenzione specifica all'elaborazione dei segnali elettrocardiografici (ECG). Sfruttando la naturale sparsità dei segnali biomedici, questa tesi sviluppa metodologie innovative per ottenere una trasmissione sicura dei dati, una ricostruzione efficiente del segnale e una diagnosi accurata. I framework proposti bilanciano avanzamenti teorici e implementazioni pratiche, colmando le lacune nelle attuali tecnologie indossabili e nei sistemi di monitoraggio remoto.

I contributi principali includono la progettazione di un framework crittografico a doppio strato che combina il compressive sensing per una trasmissione sicura ed energeticamente efficiente dei dati ECG, validata su piattaforme hardware a basso consumo. La tesi introduce inoltre una matrice di sensing basata sui dati e dizionari sovracompleti specifici per i segnali ECG, consentendo alti rapporti di compressione senza compromettere la qualità della ricostruzione. Inoltre, integra tecniche di deep learning nel framework CS, proponendo modelli basati su autoencoder per un rilevamento e una ricostruzione efficienti. Questi modelli si adattano dinamicamente ai dati, raggiungendo elevate prestazioni in termini di accuratezza ed efficienza computazionale rispetto agli studi allo stato dell'arte.

Andando oltre la tecnica tradizionale di ricostruzione del segnale, la ricerca sviluppa un framework di compressed learning che classifica direttamente i segnali ECG nel dominio compresso, evitando il passaggio di ricostruzione. Questo approccio riduce significativamente il carico computazionale mantenendo accuratezza diagnostica, con validazioni hardware che ne dimostrano la fattibilità per applicazioni nel mondo reale.

I risultati evidenziano il potenziale trasformativo del compressive sensing e del compressed learning nel consentire soluzioni sanitarie scalabili, sicure ed energeticamente efficienti. Questa tesi pone dunque una solida base per le tecnologie

indossabili di nuova generazione e i sistemi di monitoraggio remoto, contribuendo a migliorare l'accessibilità sanitaria e l'affidabilità diagnostica. Le future direzioni di ricerca includono l'estensione di queste metodologie ad altri segnali fisiologici, l'ottimizzazione delle implementazioni hardware e l'esplorazione di sistemi multimodali di monitoraggio della salute.

Contents

List of Figures

List of Tables

List of Abbreviations

BANs	Body Area Networks
BDM	Block Diagonal Matrix
BPDN	Basis Pursuit De-noising
BSBL	Block Sparse Bayesian Learning
BTM	Binary Thresholding Matrix
CC	Correlation Coefficient
CL	Compressed Learning
CNN	Convolutional Neural Networks
CR	Compression Ratio
CS	Compressive Sensing or Compressed Sensing
CVD	Cardiovascular Diseases
CVX	Convex Optimization
DCT	Discrete Cosine Transform
DSM	Data Drive Sensing Matrix
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EHR	Electronic Health Record
FF	Flip Flops
FPGA	Field Programmable Gate Array
FSM	Fourier Structure Matrix
GD	Gaissian Dictionary
IHT	Iterative Hard Thresholding
IoMT	Internet of Medical Things
IoT	Internet of Things
JLL	Johnson-Lindenstrauss Lemma
LDPC	Low Density Parity Check Matrix
LT	Lookup Tables

MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
NN	Neural Networks
OMP	Orthogonal Matching Purshit
PCA	Principal Component Analysis
PRD	Percentage RMS (Root Mean Square) Difference
QS	Quality Score
rbio	Reverse Biorthogonal Wavelets
RBMB	Random Bernoulli Matrix with Binary elements
RBMS	Random Bernoulli Matrix with Signed elements
RGM	Random Gaussian Matrix
RIP	Redistricted Isomatry Property
RX	Receiver
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TX	Transmitter
VHDL	VHSIC (Very High-Speed Integrated Circuit) Hardware Description Language
WBAN	Wireless Body Area Network
WDD	Weighted Diagnostic Distortion
WEDD	Wavelet Energy Diagnostic Distortion

Chapter 1

Introduction

1.1 Motivation

The growing prevalence of cardiovascular diseases (CVDs) has positioned them as the leading cause of mortality worldwide, accounting for approximately 17.9 million deaths annually. Early diagnosis and continuous monitoring of cardiac health are pivotal in mitigating this global health crisis. Among the diagnostic tools available, the electrocardiogram (ECG) has proven to be an indispensable method for detecting a wide range of heart-related anomalies, including arrhythmias and ischemic conditions. Despite its diagnostic significance, traditional ECG acquisition and processing methods face several limitations, particularly in the context of modern healthcare systems that emphasize portability, real-time monitoring, and accessibility.

The rapid development of wearable health technologies and portable medical devices has introduced new possibilities for healthcare delivery. These advancements enable continuous monitoring of physiological signals, allowing for early detection of abnormalities and improved management of chronic conditions. However, such systems are constrained by limited computational resources, energy efficiency, and data storage capacities. ECG signals, typically characterized by high sampling rates and substantial data volumes, pose significant challenges for these resource-constrained environments. These challenges motivate the exploration of innovative solutions that ensure high diagnostic accuracy while minimizing resource consumption.

Compressive sensing (CS) has emerged as a transformative framework capable of addressing these challenges. Unlike traditional signal acquisition techniques that adhere to the Nyquist-Shannon sampling theorem, CS leverages the sparsity inherent in many natural and biomedical signals to enable simultaneous acquisition and compression. By reducing the number of required measurements without compromising signal integrity, CS holds the potential to significantly enhance the efficiency of data acquisition, transmission, and storage in ECG monitoring systems. This capability is particularly valuable for wearable devices and remote health monitoring applications, where energy conservation and bandwidth limitations are critical considerations.

The motivation for this thesis is grounded in the need to develop efficient, scalable, and secure frameworks for ECG signal processing using compressive sensing. Traditional approaches to ECG signal acquisition often involve high sampling rates and subsequent compression, resulting in significant energy consumption and transmission overheads. Moreover, the reliance on computationally intensive reconstruction algorithms and storage-heavy sensing matrices presents further obstacles to real-time applications. These limitations underscore the need for a comprehensive framework that integrates efficient sensing matrices, sparse representations, and robust reconstruction methods tailored specifically for ECG signals.

A key aspect of this work is the focus on designing sensing matrices that align with the practical requirements of real-world healthcare systems. Random sensing matrices, while theoretically robust, are often computationally expensive and challenging to implement in hardware. Structured and data-driven sensing matrices, on the other hand, offer the potential for improved performance and resource efficiency. This thesis aims to explore the design and optimization of such matrices, with a particular emphasis on binary matrices that are well-suited for hardware implementation. By addressing the limitations of traditional sensing matrices, this work seeks to advance the state-of-the-art in ECG signal acquisition and processing.

Another motivating factor is the integration of machine learning techniques to enhance the capabilities of compressive sensing. Data-driven approaches, including dictionary learning and adaptive sensing matrix design, have demonstrated their potential to improve signal sparsity and reconstruction accuracy. By incorporating these advancements, this research aims to bridge the gap between compressive sensing and modern machine learning, enabling more efficient and accurate ECG signal processing. The combination of compressive sensing and machine learning also facilitates direct feature extraction from compressed data, eliminating the need for computationally expensive reconstruction processes in certain applications.

In addition to its technical significance, this research is driven by its potential societal impact. Cardiovascular diseases disproportionately affect individuals in low-resource settings, where access to advanced diagnostic tools is often limited. By enabling the development of low-cost, energy-efficient, and portable ECG monitoring systems, this work contributes to improving healthcare accessibility and equity. Real-time ECG monitoring systems powered by compressive sensing can facilitate the detection of cardiac abnormalities, reducing the burden on healthcare systems and enhancing patient outcomes.

Furthermore, the inherent randomness of the compressive sensing process introduces an additional layer of data security, which is increasingly important in the context of sensitive medical information. This feature has motivated researchers to explore the use of compressive sensing not only for efficient data acquisition but also as a cryptographic tool for secure transmission and storage of biomedical signals. This thesis builds on these insights to develop a secure framework for ECG signal processing, addressing the growing concerns around patient data privacy in the digital age.

Overall, the motivation for this research is rooted in the convergence of technical challenges, research opportunities, and societal needs. By addressing the limitations of traditional ECG signal processing methods and leveraging the potential of compressive sensing, this work aims to contribute to the development of next-generation healthcare solutions. These solutions promise to be efficient, secure, and impactful, paving the way for advancements in wearable health technologies and remote monitoring systems. This research not only advances the field of biomedical signal processing but also aligns with the broader goal of improving global healthcare access and quality.

Thesis Outline

This thesis is structured to systematically address the challenges and opportunities in applying compressive sensing (CS) and compressed learning to ECG signal processing. Each chapter builds upon the previous, creating a coherent narrative that transitions from foundational concepts to practical implementations. Below is an outline of the thesis:

Chapter 1: Introduction

The introductory chapter sets the stage for the research by outlining the motivation and significance of efficient ECG signal processing in modern healthcare. It highlights the limitations of traditional methods, such as the high energy and computational demands of wearable devices, and introduces compressive sensing as a powerful alternative. This chapter provides a roadmap of the thesis structure, offering the reader a clear understanding of the study's scope and contributions.

Chapter 2: Compressive Sensing Fundamentals and Related Work

This chapter provides a detailed review of the theoretical principles underlying compressive sensing, including sparsity, the Restricted Isometry Property (RIP), and incoherence. These concepts are critically analyzed in the context of ECG signal processing, laying the groundwork for the novel methodologies proposed later. The chapter also surveys existing applications of CS in ECG signals, identifying gaps in the current state of the art. This review serves to position the thesis contributions within a broader scope.

Chapter 3: Compressive Sensing for Secure and Energy-Efficient ECG Signals

Chapter 3 focuses on addressing two important challenges in wearable healthcare systems: data security and energy efficiency. It introduces a novel cryptographic framework that integrates compressive sensing with column permutation, creating a dual-layer encryption mechanism. This approach enhances the security of sensitive patient data while maintaining the integrity of signal reconstruction. Additionally, this chapter evaluates the performance of the proposed framework under different compression ratios and demonstrates its energy efficiency through hardware implementations on microcontrollers and FPGA.

Chapter 4: Data-Driven Sensing Matrix and Dictionary Design for ECG Signals

The fourth chapter explores the development of sensing matrices and sparse representations to optimize ECG signal. A data-driven sensing matrix is proposed, leveraging the unique characteristics of ECG signals to improve compression and reconstruction quality. This chapter also introduces overcomplete sparse dictionaries, including Gaussian and K-SVD dictionaries, which enhance the sparsity of signal representation. A novel binary thresholding technique is presented to convert the data-driven sensing matrix into a binary format, significantly reducing computational overhead and making it suitable for hardware deployment.Experimental results validate these contributions, demonstrating their effectiveness and scalability.

Chapter 5: Deep Learning Approaches for Compressive Sensing

In this chapter, the thesis transitions into the integration of deep learning techniques within the CS framework. It proposes end-to-end deep learning models, particularly autoencoder-based architectures, to simultaneously perform sensing and reconstruction of ECG signals. These models adapt dynamically to the data during training, achieving high reconstruction fidelity even at aggressive compression ratios. The chapter compares these deep learning approaches with traditional reconstruction methods, highlighting their advantages in terms of accuracy and computational efficiency. This work establishes a new benchmark for applying machine learning to compressive sensing.

Chapter 6: Learning on Compressed ECG Measurements

Theis chapter focuses on utilizing compressed ECG data directly for diagnostic and identification tasks. A compressed learning framework is proposed, enabling the extraction of diagnostic features from compressed data without requiring full signal reconstruction. This approach significantly reduces computational overhead and is validated for three different types of signals. The chapter also explores real-world implementations of the framework on hardware platforms, such as microcontroller, showcasing its practicality for wearable devices. A comprehensive comparison with existing methods emphasizes the superiority of the proposed approach in terms of energy efficiency, accuracy, and scalability.

Chapter 7: Conclusion and Future Work

The concluding chapter synthesizes the key findings and contributions of the research. It reflects on how the proposed frameworks advance the field of ECG signal processing, addressing critical challenges in data compression, secure transmission, and real-time classification. The chapter also outlines potential avenues for future exploration, including the extension of these methodologies to other physiological signals and the integration of multimodal health monitoring systems.

Overall, this thesis bridges the gap between theoretical advancements and practical applications in compressive sensing and compressed learning. By proposing innovative techniques for efficient, secure, and scalable ECG signal processing, it contributes to the development of next-generation wearable and remote healthcare solutions.

1.2 Summary of Contributions

This thesis contributes significantly to the fields of compressive sensing and compressed learning for ECG signal processing, addressing key challenges in data volume reduction, energy efficiency, security and diagnostic reliability in health monitoring systems. The primary contributions of this work are summarized as follows:

- A systematic review of compressive sensing for physiological signals, including ECG, EEG, EMG, and GSR, was conducted. This review identified critical challenges in the current state-of-the-art CS technologies for healthcare applications [P4 and P10].
- The sparsity of ECG signals in various standard dictionaries was analyzed, and over-complete dictionaries were proposed to enhance sparse representations of ECG signals in compressive sensing applications [P2 and P5[.
- To address the limitations of existing CS-based encryption systems, a dual-layer encryption approach was developed, integrating column permutation with CS encryption. The proposed system was successfully implemented and validated on hardware [P1 and P7].
- A data-driven sensing matrix was proposed to extract critical features from ECG signals at high compression ratios. The matrix's performance was compared with state-of-the-art sensing matrices. Additionally, a novel thresholding technique was introduced to convert the sensing matrix into a binary format for enhanced hardware efficiency, and its implementation was validated on hardware platforms [Chapter 4].
- Lightweight, non-iterative machine learning models were developed for compressed signal reconstruction. These models were validated across various sensing matrices and benchmarked against state-of-the-art reconstruction algorithms. Their efficiency was further demonstrated through hardware implementation and resource utilization analysis [Chapter 5].
- Lightweight end-to-end sensing and reconstruction models were introduced using an autoencoder-based approach. These models were implemented on hardware, and their resource consumption was evaluated to ensure practical feasibility [Chapter 5].
- A compressed learning approach was proposed to extract features directly from compressed data, bypassing traditional reconstruction stages. The accuracy of the proposed model was rigorously compared with standard classification methods, demonstrating its potential for real-time healthcare applications [P6 P8 and Chapter 6].

.

Chapter 2

Compressive Sensing Fundamentals and Related Work

2.1 Introduction

In today's world, the widespread use of digital signals and images has become integral to many aspects of everyday life. From entertainment to healthcare, digital data plays a crucial role in modern society. Healthcare data has become one of the growing sectors, in the digital realm producing an estimated 2,314 exabytes of data every year. Recent study [1] shows the 30% of global data is generate by the healthcare sector. By the year 2025 the rate at which healthcare data grows annually is projected to reach 36% surpassing that of manufacturing by 6% financial services by 10% and media and entertainment by 11%. It is estimated that individuals will engage in over 1,400 interactions per day by the end of this year and nearly 5,000 within next five years with a significant portion being related to healthcare. This surge in data is primarily fuelled by the increasing adoption of electronic health records (EHRs) medical imaging technologies, genomic sequencing and the rising popularity of health monitoring devices specially physiological signals. The widespread use of digital devices and internet platforms has resulted in an exponential rise in data volume highlighting the need for effective data management strategies. Conventional techniques for data compression typically involve gathering data in its original form before compressing it for storage or transmission purposes. However, this conventional method may prove inefficient and costly regarding memory usage, processing power requirements and time constraints. Moreover, owing to the quantity and diverse nature of healthcare related data available today it becomes a prime target, for potential cybersecurity threats. Securing and protecting information is a crucial task that demands the use of strong encryption, access controls and authentication methods to keep sensitive data safe. Compressive sensing (CS) also called compressed sensing or compressed sampling introduces an approach by directly capturing signals in a compressed form at sub Nyquist sampling rate, and reconstruct the signal from compressed measurements. Moreover, CS ensures the security of healthcare data. This method reduces data volume from the initial data acquisition stage making it highly appealing in today's digital age. Despite being a recent subject compressive sensing has attracted significant interest from researchers for its potential to reduce data volume and related expenses.

This chapter provides an in-depth overview of the framework of compressive sensing emphasizing its principles, key features and practical applications. The focus is on it's application to physiological signals.

2.2 Digital Signal Acquisitions and Representation

In the modern age of digital and internet revolutions, data in the form of signals are increasing in every aspect of our daily lives. From communications to entertainment, and from healthcare to scientific research, the role of signal acquisition and processing is pivotal. These technologies lies on the theoretical framework provided by the Nyquist–Shannon sampling theorem, which has been instrumental in the field of digital signal processing. This theorem asserts that a continuous-time, band-limited signal can be completely reconstructed from its samples if the sampling rate is at least twice the highest frequency present in the signal. This minimum sampling rate is known as the Nyquist rate [2, 3].

While the Nyquist–Shannon theorem has been foundational in transforming analog signals into digital format, leading to significant technological advancements, it also introduces practical challenges. Signals sampled at the Nyquist rate often result in large datasets that are difficult to store, process, and transmit. In resourse constraint environment such as wearable sensors and other wireless devices, managing these voluminous signals becomes increasingly difficult.

One traditional approach to addressing these challenges is through signal compression, which aims to reduce the size of the signal without significant loss of information. Transform coding is a widely adopted compression technique that seeks to represent a signal in a basis where it is sparse. A sparse signal is one where most of its components are zero or near-zero, allowing the entire signal to be represented by a few significant non-zero coefficients. In this context, a signal of length n can be efficiently described by k non-zero coefficients, where k is much smaller than n [4, 5]. In addition, it must be noted that such a paradigm involves three consecutive stages (i.e., sampling at Nyquist's frequency, compressing to reduce the amount of information to be transmitted, and, eventually, decompressing at the receiving node), which significantly slower the acquisition process.

The concept of compressive sensing, introduced by Candes, Romberg, Tao, and Donoho, revolutionizes traditional signal processing by allowing signals to be sampled below the Nyquist rate, provided they have a sparse representation in some basis (e.g., Fourier or wavelet transform) [6–9]. Compressive sensing exploits the inherent sparsity of many natural signals to reduce the number of samples needed for accurate reconstruction. This approach significantly reduces the amount of data to be acquired and processed at the sensor level, thus addressing the limitations of traditional sampling methods. To this aim, it uses mathematical tools, such as the L1 norm at the receiving node to represent or reconstruct the original sparse signal from only a few samples.

In recent years, CS has gained the attention of researchers for its applicability in several fields, including power line communications [10], compressive imaging [11–13], biomedical applications [14]. ultrasound imaging [15], face recognition [16, 17], single-pixel cameras [18], wireless sensor networks [19], cognitive radio networks [20, 21], sound localization [22], audio processing [23, 24], and video processing [25, 26]. In neural engineering, CS has proven valuable in studying neurons connectivity [27], improving MRI acquisition and reconstruction [28, 29], monitoring electroencephalograms (EEG) [30], and other applications. This indicates that a small number of linear projections can better reconstruct the original signal than a large number of linear projections.

The main challenges in compressive sensing revolve around three key aspects: (i) how to effectively measure a signal compressively, (ii) how to apply this method to practical scenarios, and (iii) how to accurately reconstruct the signal from its compressed measurements.

2.2.1 Signal Vectorial Representation

CS operates fundamentally with signals in a vectorial form, it is important to understand how signals can be represented as vectors. Natural or artificially generated signals are inherently linear in nature. A signal can thus be represented as a 'vector' in a 'vector space'.

Signal Vector Definition

A signal of length n can be represented as an n-dimensional vector in the vector space $\mathbb{R}^n$, expressed as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

A common measure of the magnitude of a signal vector is its l_p norm, where p is a real number ranging from 1 to infinity; $p \in [1, \infty]$.

Definition of l_p Norm

The l_p norm of a signal vector x is defined by:

$$||x||_p := \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, & p \in [1, \infty), \\ \max_{i=1,2,\ldots,n} |x_i|, & p = \infty. \end{cases}$$

The l_p norm provides a measure of the vector's magnitude and can also be used to calculate the distance between two vectors.

l_1 Norm

The l_1 norm, also referred to as the 'Manhattan distance' or 'Taxicab norm,' is calculated by summing the absolute values of the vector's elements:

$$||x||_1 := |x_1| + |x_2| + \cdots + |x_n|.$$

This norm is analogous to navigating a grid-like path in a city, moving along the current space dimensions.

l_2 Norm

The l_2 norm, or Euclidean norm, measures the shortest distance represented by a vector, defined as:

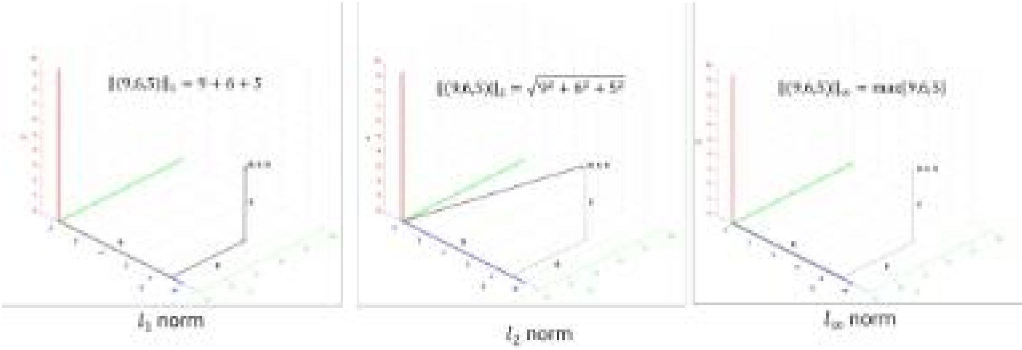

Figure 2.1: Example of l_1,l_2 and l_∞ norms

$$||x||_2 := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

This norm can also be described using the inner product in the vector space $\mathbb{R}^n$, expressed as $||x||_2 = \sqrt{\langle x, x \rangle}$.

l_∞ Norm

The l_∞ norm identifies the largest absolute value among the vector's elements. Mathematically, it is represented as:

$$||x||_\infty := \max_{i=1,2,\ldots,n} |x_i|.$$

This norm focuses on the maximum deviation, which is particularly useful in scenarios requiring the largest absolute value. Furthermore, the pictorial definition of l_1 norm l_2 norm and l_∞ norm is shown in the figure 2.1.

2.2.2 Sparse Basis or Dictionaries

In a vector space, the basis also called dictionary is a set of vectors that are linearly independent and span the entire vector space. This means that any vector in the space can be uniquely expressed as a linear combination of the basis vectors. Consider a vector space $\mathbb{R}^n$. A set of n linearly independent vectors $\{\psi_i \mid i = 1, \ldots, n\}$ forms a basis if any vector x in the vector space can be uniquely represented as:

$$x = \sum_{i=1}^{n} c_i \psi_i$$

where c_i are the coefficients. In matrix notation, if we let Ψ be an $n \times n$ matrix then:

$$x = \Psi c$$

where c is the vector of coefficients.

Orthogonal Basis

An orthogonal basis is a special type of basis where all the basis vectors are orthogonal to each other. If, in addition, all the basis vectors have unit length, the basis is said to be orthonormal.

1. **Orthogonality:** Vectors ψ_i and ψ_j are orthogonal if their inner product is zero:
$$\langle \psi_i, \psi_j \rangle = 0 \quad \text{for } i \neq j$$

2. **Orthonormality:** Vectors ψ_i are orthonormal if they are orthogonal and each vector has unit length:
$$\langle \psi_i, \psi_i \rangle = 1 \quad \text{and} \quad \langle \psi_i, \psi_j \rangle = 0 \quad \text{for } i \neq j$$

For an orthonormal basis, the matrix Ψ has the property:

$$\Psi^T \Psi = I$$

where I is the identity matrix. This implies that the coefficients can be easily computed as:
$$c = \Psi^T x$$

Overcomplete Dictionary

An overcomplete dictionary is a set of vectors that spans the vector space but includes more vectors than the dimensionality of the space. This means the dictionary contains redundant vectors, providing multiple ways to represent any given vector in the space.

1. **Redundancy:** An overcomplete dictionary has more vectors than the dimensionality of the space, i.e., if the space is $\mathbb{R}^n$, the dictionary has m vectors where $m > n$.

2. **Linear Dependence:** Unlike a basis, the vectors in an overcomplete dictionary are not linearly independent.

If Ψ is an $n \times m$ matrix representing the overcomplete dictionary with $m > n$, then any vector x in the vector space can be represented as:

$$x = \Psi c$$

where c is an m-dimensional vector. Since Ψ has more columns than rows, there are infinitely many possible solutions for c.

Overcomplete dictionaries are useful in compressive sensing and sparse representation because they provide flexibility and robustness. The redundancy allows for sparse representations, meaning that a signal can be represented with only a few non-zero coefficients in the dictionary, which is crucial for efficient data compression and reconstruction.

2.3 Sparsity and Compression

2.3.1 Sparsity

Sparsity is a key concept in compressive sensing and data compression. It refers to the property of signals where most of the information is contained in a small number of significant coefficients, while the majority of the coefficients are negligible or zero. This is particularly true for many natural signals, where a few large values capture most of the signal's energy. Consequently, a signal vector $\mathbf{x} \in \mathbb{R}^n$ is considered k-sparse if it has at most k non-zero elements:

$$\|\mathbf{x}\|_0 \leq k. \tag{2.1}$$

It should be noted that $\mathbf{x} \neq 0$. Indeed, the zero vector is outside the scope of sparsity analysis. Compressive sensing does not necessarily require the signal vector to be k-sparse in its original form; it is sufficient if it is k-sparse in some basis. For instance, a harmonic signal can have a sparse representation in the Fourier basis Ψ. Thus, if $\mathbf{s}$ is the representation of $\mathbf{x}$ in the basis Ψ, then:

$$\mathbf{x} = \Psi \mathbf{s}, \tag{2.2}$$

where $\mathbf{s}$ is a k-sparse vector. In many cases, such as Fourier or wavelet transforms, the transform basis Ψ is fixed and well-known, facilitating efficient computation and representation of $\mathbf{x}$ with a few significant coefficients $\mathbf{s}$.

2.3.2 Sparsity and Compressibility

A compressible signal is one where a few elements of the signal vector can represent the entire signal. This means the signal vector can be approximated by keeping only the significant elements and ignoring the rest, effectively compressing the signal. For a compressible signal vector $\mathbf{x}$ of length n, let $\hat{\mathbf{x}}$ be its compressed representation with the k dominant elements. The approximation error can be measured using the l_p norm as follows:

$$e_k(\mathbf{x})_p = \min_{\hat{\mathbf{x}} \in \Sigma_k} \|\mathbf{x} - \hat{\mathbf{x}}\|_p, \tag{2.3}$$

where Σ_k is the set of all k-sparse vectors. If the signal is truly k-sparse, this error is zero. This indicates that as the signal becomes closer to being k-sparse, the compression error approaches zero:

$$\mathbf{x} \in \Sigma_k \Rightarrow \forall p, e_k(\mathbf{x})_p = 0. \tag{2.4}$$

2.4 Compressive Sensing Fundamentals

Compressive sensing is a novel paradigm that combines sampling and compression into a single step. Instead of sampling at the Nyquist rate and then compressing, CS directly acquires a compressed representation of the signal using a sampling rate significantly below the Nyquist rate. This is particularly advantageous in scenarios where acquiring full-resolution data is impractical due to constraints on sampling time, storage, or energy.

The theory of compressed sensing was initially proposed by Candès and Tao [31]. This approach has proven to be an effective method for acquiring and reconstructing signals by taking advantage of the fact that physical signals are typically sparse in their original form or on a well-selected basis (e.g., Fourier or wavelet transform). By directly acquiring a compressed form of the signal in a single stage, CS employs mathematical tools like the L_1 norm to represent or reconstruct the original sparse signal from only a few samples at the receiving node.

2.4.1 Mathematical Model of Compressed Sensing

Physiological signals such as EEG, ECG, and EMG exhibit significant sparsity and high compressibility, making them well-suited for efficient representation and processing. When these signals are compressed, only a limited number of components remain active when expressed in an appropriate basis, thereby reducing the amount of data required for an accurate reconstruction. A compressible signal $x \in \mathbb{R}^n$ can be represented as a sparse matrix $s \in \mathbb{R}^n$, predominantly composed of zeros, in a transform basis such as the Fourier or wavelet basis $\psi \in \mathbb{R}^{n \times n}$. When the vector s contains K nonzero elements, the signal is referred to as K-sparse in the basis ψ. Mathematically, this relationship is expressed as:

$$x = \psi s = \sum_{i=1}^{N} \psi_i s_i \tag{2.5}$$

This formulation highlights the fundamental principles underlying sparse signal representation, which are pivotal for efficient signal compression and reconstruction in various applications.

CS utilized the sparsity of a signal in a transform basis to achieve complete signal reconstruction from a surprisingly small number of measurements. If a signal $\mathbf{x}$ is K-sparse in the basis Ψ, then instead of directly measuring $\mathbf{x}$ using n measurements and subsequently compressing it, one can obtain significantly fewer randomly chosen or compressed measurements and solve for the nonzero elements of $\mathbf{s}$ in the transformed system. The measurements $\mathbf{y} \in \mathbb{R}^m$, where $K < m \ll n$, are given by:

$$\mathbf{y} = \mathbf{\Phi x} \tag{2.6}$$

The Sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ represents a set of m linear measurements on the state $\mathbf{x}$. The selection of the measurement matrix $\mathbf{\Phi}$ is crucial in compressed sensing. Typically, measurements may consist of random projections of the state, in which case the entries of $\mathbf{\Phi}$ are Gaussian or Bernoulli distributed random variables.

Given the knowledge of the sparse vector $\mathbf{s}$, the signal $\mathbf{x}$ can be reconstructed from the compressed measurements. The objective of compressed sensing is to identify the sparsest vector $\mathbf{s}$ that is consistent with the measurements $\mathbf{y}$. Eq. 2.6 can be written as Eq. 2.7 and also shown in Fig. 2.2.

$$\mathbf{y} = \mathbf{\Phi} \Psi \mathbf{s} = \Theta \mathbf{s}. \tag{2.7}$$

Where, $\Theta = \mathbf{\Phi}\Psi$ is $m \times n$ reconstruction matrix. This system of equations is underdetermined since there are infinitely many solutions for $\mathbf{s}$. The sparsest solution $\hat{\mathbf{s}}$ is determined by solving the following optimization problem:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\Psi\mathbf{s}, \tag{2.8}$$

where $\|\cdot\|_0$ denotes the l_0 pseudo-norm, representing the number of nonzero entries. This optimization problem is non-convex, and generally, the solution can only be found through a brute-force search that is combinatorial in n and K. This search becomes intractable for even moderately large n and K.

Fortunately, under certain conditions on the measurement matrix $\mathbf{\Phi}$, it is feasible to relax the optimization problem to a convex l_1-minimization problem [8, 9]:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\Psi\mathbf{s}, \tag{2.9}$$

where $\|\cdot\|_1$ is the l_1 norm, defined as:

$$\|\mathbf{s}\|_1 = \sum_{k=1}^{n} |s_k|. \tag{2.10}$$

The l_1 norm, also known as the taxicab or Manhattan norm, represents the distance between two points on a rectangular grid.

Certain conditions must be met for the l_1-minimization to converge with high probability to the sparsest solution [6, 7]. These conditions include:

1. The measurement matrix $\mathbf{\Phi}$ must be incoherent with respect to the sparsifying basis Ψ, meaning that the rows of $\mathbf{\Phi}$ are not correlated with the columns of Ψ. 2. The number of measurements m must be sufficiently large, on the order of:

$$m \approx O(K \log(n/K)) \approx k_1 K \log(n/K). \tag{2.11}$$

The constant multiplier k_1 depends on the level of incoherence between $\mathbf{\Phi}$ and Ψ. These conditions ensure that the matrix $\mathbf{\Phi}\Psi$ acts as a unitary transformation on K-sparse vectors $\mathbf{s}$, preserving relative distances between vectors and enabling almost certain signal reconstruction with l_1 convex minimization. This is precisely formulated in terms of the Restricted Isometry Property (RIP) in section 2.5.1.

The Shannon-Nyquist sampling theorem states that perfect signal recovery requires sampling at twice the rate of the highest frequency present [32, 33]. However, this result only provides a strict bound on the required sampling rate for signals with broadband frequency content. Since an uncompressed signal will generally be sparse in a transform basis, the Shannon-Nyquist theorem may be relaxed, allowing signal reconstruction with significantly fewer measurements than given by the Nyquist rate. However, even though the number of measurements may be reduced, compressed sensing still relies on precise timing of the measurements. Moreover, signal recovery via compressed sensing is not guaranteed but is possible with high probability, making it primarily a statistical theory. The probability of successful recovery becomes astronomically large for moderately sized problems.

In addition to the l_1-minimization, alternative approaches based on greedy algorithms [34–40] determine the sparse solution through an iterative matching pursuit problem. For instance, the Compressed Sensing Matching Pursuit (CoSaMP) [41] is computationally efficient, easy to implement, and freely available.

When the measurements $\mathbf{y}$ have additive noise, such as white noise of magnitude ϵ, robust variants of the l_1-minimization are employed:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{subject to} \quad \|\mathbf{\Phi}\Psi\mathbf{s} - \mathbf{y}\|_2 < \epsilon. \tag{2.12}$$

A related convex optimization problem is:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{\Phi}\Psi\mathbf{s} - \mathbf{y}\|_2 + \lambda\|\mathbf{s}\|_1, \tag{2.13}$$

where $\lambda \geq 0$ is a parameter that weights the importance of sparsity.

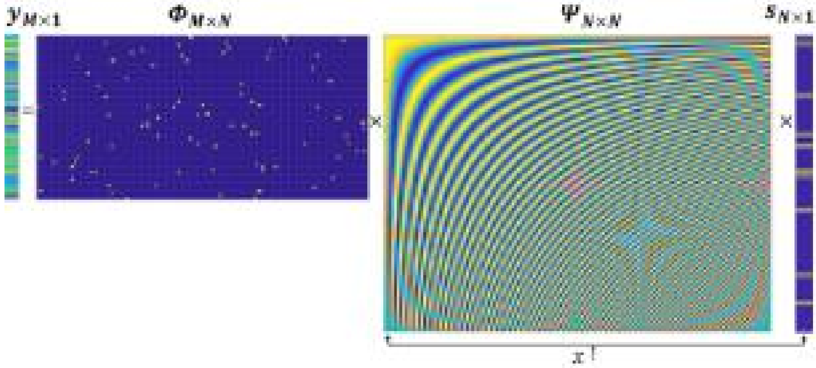

Figure 2.2: Compressive Sensing Framework

2.5 Essential Conditions of Compressive Sensing Matrices

The success of CS largely depends on two critical conditions: the Restricted Isometry Property (RIP) and the incoherence of the measurement matrices. These conditions ensure that the measurements capture sufficient information to allow for accurate signal reconstruction. This section explores these essential conditions in detail, highlighting their importance and the underlying mathematical principles.

2.5.1 The Restricted Isometry Property (RIP)

The Restricted Isometry Property (RIP) is a fundamental criterion for the effectiveness of a measurement matrix in compressed sensing. A matrix $\Phi\Psi$ is said to satisfy the RIP of order K if there exists a constant $\delta_K \in (0, 1)$ such that for all K-sparse vectors $\mathbf{s}$:

$$(1 - \delta_K)\|\mathbf{s}\|_2^2 \leq \|\boldsymbol{\Phi\Psi}\mathbf{s}\|_2^2 \leq (1 + \delta_K)\|\mathbf{s}\|_2^2.$$

Here, the constant δ_K is known as the restricted isometry constant [42]. A smaller δ_K implies that the matrices $\boldsymbol{\Phi\Psi}$ acts nearly as an isometry on K-sparse vectors, preserving the Euclidean norm of these vectors. This property ensures that the distances between sparse signals are approximately preserved during the measurement process, which is crucial for accurate signal reconstruction.

In practice, directly computing δ_K is challenging, and thus, statistical properties of δ_K for families of measurement matrices are often analyzed instead. Increasing the number of measurements generally decreases δ_K, enhancing the isometric properties of $\boldsymbol{\Phi\Psi}$. When there are sufficiently many incoherent measurements, it becomes possible to accurately recover the K nonzero elements of an n-length vector $\mathbf{s}$ [8, 9].

2.5.2 Incoherence and Measurement Matrices

Incoherence is another pivotal concept in compressed sensing, referring to the lack of correlation between the measurement matrix $\boldsymbol{\Phi}$ and the sparsifying basis $\boldsymbol{\Psi}$. A measurement matrix $\boldsymbol{\Phi}$ is said to be incoherent with respect to the basis $\boldsymbol{\Psi}$ if the inner products between the rows of $\boldsymbol{\Phi}$ and the columns of $\boldsymbol{\Psi}$ are uniformly small.

Mathematically, for a measurement matrix $\mathbf{C}$ and sparsifying basis $\mathbf{\Psi}$, the coherence μ is defined as:

$$\mu(\mathbf{C}, \mathbf{\Psi}) = \max_{i,j} |\langle \mathbf{c}_i, \psi_j \rangle|.$$

Low coherence is essential as it ensures that the measurement process spreads the information of each sparse coefficient across all measurements, making the recovery process robust. Random matrices such as Gaussian and Bernoulli matrices are known to satisfy the RIP with high probability for a wide range of bases $\mathbf{\Psi}$. These random matrices exhibit low coherence, making them suitable for compressed sensing applications.

In numerous engineering applications, it is beneficial to represent the signal x in a generic basis, such as the Fourier or Wavelet basis. A significant advantage of this approach is that single-point measurements are incoherent with respect to these bases, thereby exciting a broadband frequency response. Sampling at random point locations is particularly appealing in scenarios where individual measurements are costly, such as ocean monitoring. Examples of random measurement matrices include single-pixel, Gaussian, Bernoulli, and sparse random matrices.

A particularly useful transform basis for compressed sensing is derived from the Singular Value Decomposition (SVD), resulting in a tailored basis in which the data is optimally sparse [43–47]. Utilizing a truncated SVD basis can lead to more efficient signal recovery from a reduced number of measurements. Significant progress has been made in developing compressed SVD and Principal Component Analysis (PCA) techniques based on the Johnson-Lindenstrauss (JL) lemma [48–51]. The JL lemma is closely related to the Restricted Isometry Property (RIP), which indicates when it is feasible to embed high-dimensional vectors in a low-dimensional space while preserving their spectral properties.

2.6 CS Framework for ECG Signals

In CS, compressed measurements are directly acquired from the signal of interest (ECG signal), rather than following the traditional approach: 1) sampling the signal at the Nyquist rate; 2) compressing it by computing the transform coefficients to retain the larger coefficients; and 3) discarding the smaller ones for transmission or storage. With growing advancements in healthcare electronics, ECG signals represent a promising approach for advanced diagnosis and treatment of cardiovascular diseases. For ambulatory health monitoring, wireless devices are commonly used, with batteries serving as the primary energy source. In practice, acquiring, processing, and transmitting significant amounts of data over long periods consume substantial power and storage space. Therefore, efficient data acquisition and compression techniques are crucial. CS techniques for sensing or compressing ECG signals can significantly reduce battery consumption, storage space requirements, and the amount of data to be transmitted. Generally, the digital CS approach applied to ECG signals is divided into four steps, also shown in the figure 2.3:

1. **CS Encoder:** After ADC, the ECG signals can be compressed or encoded by multiplying with sensing matrix which collect the fewer measurements depending on the compression ratio.

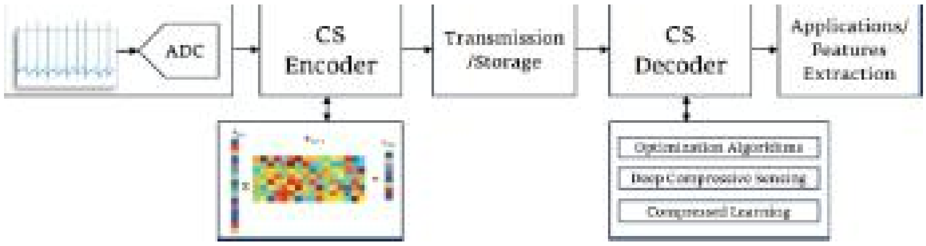

Figure 2.3: CS for Physiological Signals Block Diagram

2. **Transmission or Storage:** The compressed measurements can be transmitted or stored depending on the applications. Due to the reduced data size, the transmission requires less power and bandwidth, and storage requires less memory.

3. **CS Decoder:** The original ECG signals can be reconstructed from the compressed measurements using traditional optimization algorithms (Convex Optimization, Greedy algorithms, thresholding algorithms etc.) or Deep Compressed Sensing (Using deep learning and Neural Network). Additionally Compressed Learning can direct extract the features from compressed measurements).

4. **Feature Extraction/Applications:** After reconstruction, relevant features are extracted from the signal for further analysis, such as detecting abnormalities or monitoring vital signs.

Many research works on CS encoder and decoder to improve the compressed measurement and reconstruction process focusing on specific applications. For example, some studies have optimized the sensor placement and sampling strategy to maximize the efficiency of signal acquisition. Others have developed advanced reconstruction algorithms to improve the accuracy and speed of signal recovery. Additionally, research has been conducted on efficient feature extraction techniques for specific physiological signals. Advancements in CS for ECG signals have significant implications for the development of portable and wearable health monitoring systems. These systems can provide continuous monitoring, early detection of medical conditions, and personalized healthcare, thereby improving patient outcomes and reducing healthcare costs. Furthermore, we will discuss every block of the CS framework for ECG signals in details;

Cardiovascular diseases (CVDs) are the world's leading cause of mortality, claiming approximately 17.9 million lives annually [52]. The continuous measurement of ECG signals is essential for detecting various cardiac abnormalities, including arrhythmias [53]. For body area networks (BANs), low energy consumption is essential [54]. To overcome this problem, CS is the only technique that can simultaneously compress and sense the ECG signal. It has been observed previously that sparse signals are highly redundant [55]. For example, there is no information in ECG signals in the isoelectric region (region of low activity), while the PQRST region (high activity region) contains important information, as shown in figure 2.4 [56]. It is possible to reduce the amount of data sampled by applying CS to ECG signals, reducing energy consumption in wearable devices by reducing data transmission. The use of CS in ECG expands to a range of applications, for instance, detection and classification of fetal and maternal

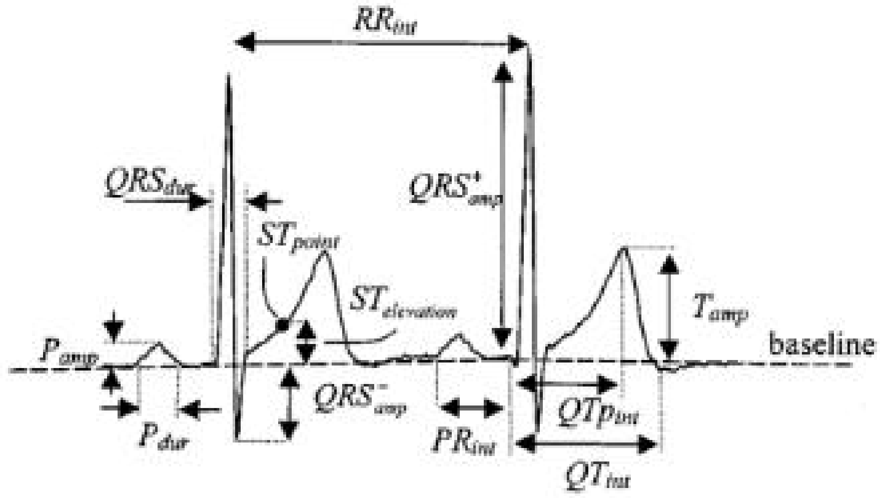

Figure 2.4: ECG morphology

beats [57], arrhythmia [57], QRS detection [58], telecardiology application [59], WBAN gateway [60], and heart rate estimation [61]. The CS theory can facilitate feature extraction from the ECG signal in several ways.

ECG Signal

Electrocardiography (ECG) records the heart's electrical activity over time using electrodes or leads. These leads capture the electrical changes due to the depolarization and repolarization of heart muscles during each heartbeat [62]. This section covers the recording of ECG signals using the standard 12-lead system, the associated noise during ECG acquisition, the morphology of the ECG signals, and the identification of heart-related diseases through ECG analysis.

ECG Leads: The spatial information of the heart's electrical activity is obtained using the standard 12-lead system, which is divided into limb and precordial (chest) leads, placed in three orthogonal directions (right to left, superior to inferior, and anterior to posterior) on the body. Limb and chest leads record the potential difference across the frontal and horizontal planes, respectively. The six limb leads include three bipolar (lead I, lead II, and lead III) and three unipolar augmented leads (aVR, aVL, and aVF). The six unipolar precordial leads are V1, V2, V3, V4, V5, and V6. Detailed descriptions and positions of the 12 leads are presented in figure 2.5 [63], illustrating the relationship between the ECG and heart orientation.

ECG Signal Morphology: The heart's activity and condition are reflected in the morphology of the ECG signal, characterized by various peaks and their durations [64]. The primary components of the ECG are the P-wave, QRS complex, T-wave, and the intervals between them. The P-wave results from the depolarization of the right and left atria. The QRS complex represents the simultaneous depolarization of the right and left ventricles. The T-wave arises from ventricular repolarization, while the U-wave follows ventricular depolarization. The intervals between these waves provide diagnostic information about heart diseases. The PR interval measures the time delay from the onset of atrial depolarization (P-wave) to the onset of ventricular depolarization (QRS complex).

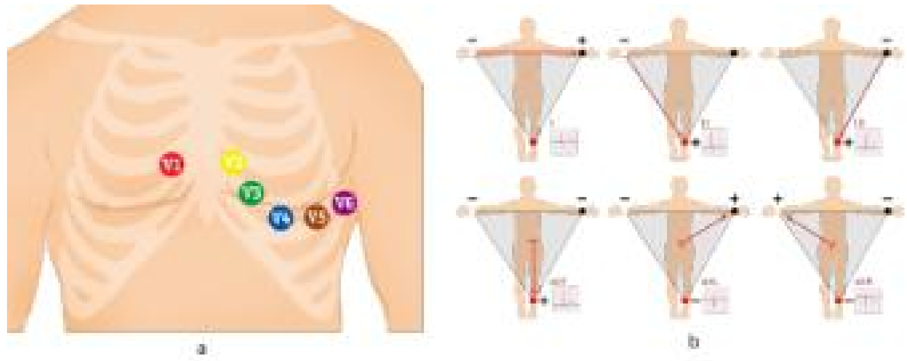

Figure 2.5: (a):Chest (Precordial) Electrodes and Placement (b): Vertical plane (Frontal Leads)

The QRS duration reflects ventricular muscle depolarization, and the QT interval encompasses the duration of ventricular depolarization and repolarization. The RR and PP intervals correspond to the durations of the ventricular and atrial cycles, respectively. Figure 2.4 illustrates the morphological characteristics of a normal ECG signal and the duration of its various waveforms.

ECG signals are essential for detecting a variety of heart-related conditions. Some of the diseases and conditions that can be identified through ECG signals include: Arrhythmias, Myocardial Infarction (Heart Attack), Ischemia, Heart Enlargement (Hypertrophy), Electrolyte Imbalance, Pericarditis, and Conduction Blocks.

2.6.1 CS Acquisition Model

Let x be the signal of concern (ECG signal) which is sparse in the domain ψ. Instead of directly measuring x with n measurements and then compressing them, it is possible to gather a smaller number of random measurements or compressive measurements directly. Mathematically, compressed measurements y are given by the following equation:

$$y = \Phi x$$

where $x \in \mathbb{R}^n$ is the signal of interest (input signal), $y \in \mathbb{R}^m$ are the compressed measurements, with $K \leq m \ll n$, and $\Phi \in \mathbb{R}^{m \times n}$ is the measurement matrix that represents m linear measurements on the signal x. Choosing the measurement matrix is critically essential and will be discussed in detail. Additionally, the signal acquisition process is shown in figure 2.6.

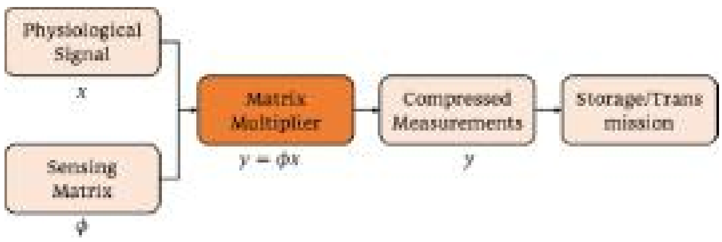

Figure 2.6: Compressive Sensing Acquisition Model

The measurement process in CS involves sampling the most representative elements of a signal using a measurement matrix. The choice of the measurement matrix is crucial, as it directly impacts both accuracy and processing time. This is because different measurement matrices require varying numbers of measurements, which significantly influence performance metrics. Consequently, designing accurate and efficient measurement matrices is a critical area of research.

Over the past decade, numerous measurement matrices have been proposed, as summarized in Figure 2.7. Measurement matrices in CS are often derived from random distributions, including both unstructured matrices (e.g., Bernoulli, Gaussian, and uniform) and structured ones (e.g., Fourier and Hadamard matrices). Unstructured random matrices, while possessing properties like orthogonality, incoherence with the signal basis, and adherence to the Restricted Isometry Property (RIP), face practical limitations. Specifically, they cannot be stored or regenerated at the receiving end, necessitating their transmission alongside compressed measurements an impractical approach for real-world applications. Moreover, random matrices generally require a larger number of measurements, which increases computational overhead.

To address these limitations, structured random matrices have been introduced. These matrices are easier to generate, require less storage, and reduce transmission overhead. However, their primary drawback is a trade-off between reduced universality and accelerated recovery procedures. As a result, researchers have shifted focus toward deterministic and semi-deterministic matrices.

Deterministic matrices are designed to meet RIP and exhibit low mutual coherence, making them suitable for various practical applications. These matrices can be further classified into semi-deterministic and fully deterministic types. Semi-deterministic matrices offer faster decoding and are versatile across multiple use cases. Fully deterministic matrices, on the other hand, are constructed based on strict adherence to mutual coherence and RIP, ensuring robust performance. Several studies [65–68] have analyzed the performance of these matrices and highlighted their suitability for applications such as physiological signal acquisition.

2.6.2 CS Reconstruction Model

CS reconstruction can be categorized into three different approaches. Traditional approaches use optimization algorithms to find the sparse solution of the signal from the compressed measurements. For this approach, the acquisition model must adhere to essential CS conditions such as the Restricted Isometry Property (RIP) and incoherence for proper reconstruction.

Deep Compressive Sensing employs deep learning or machine learning algorithms to reconstruct the original signal from compressed measurements, without the necessity to follow the essential conditions of CS.

Compressed Learning, also known as reconstruction-free CS, allows for the direct extraction of important features from compressed measurements, bypassing the need for reconstruction. The accuracy of feature extraction in this approach depends on the sensing matrix used for compression and acquisition.

Traditional Approaches

The reconstruction matrix θ and the compressive measurement vector y are the input to the reconstruction algorithm, as shown in figure 2.8. The original signal can be retrieved from compressive measurements by solving 2.6.1 or 2.7,

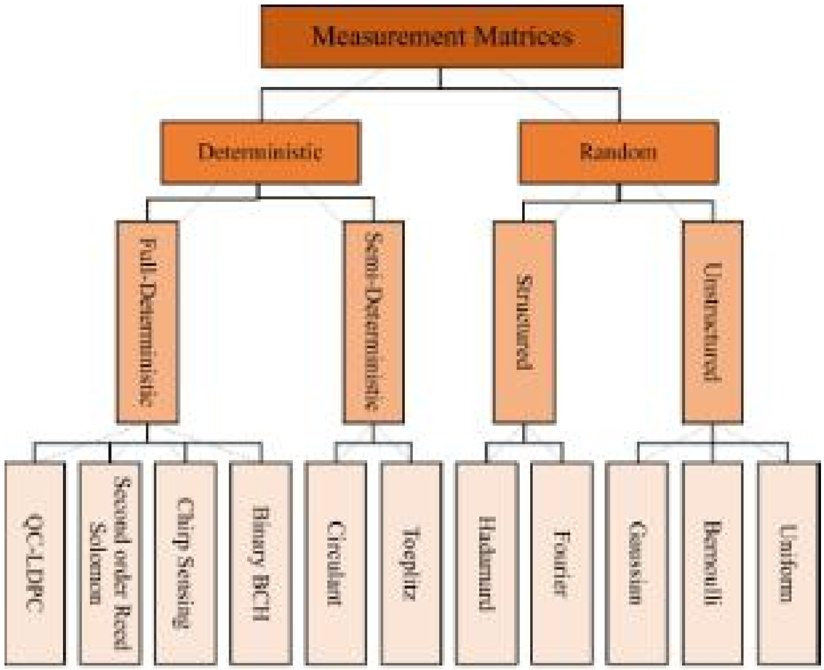

Figure 2.7: Sensing Matrices

an underdetermined linear equation system. The sparse vector s is consistent with the measurement vector y. Thus, CS has the primary objective of finding the sparsest vector $\hat{s}$ that satisfies the following conditions in (2.8), where $\|\cdot\|_0$ is the ℓ_0-pseudonorm that gives the number of entries that are not zero [69] and $\hat{s}$ is the estimate of s. Searching for the solution or optimization in (2.8) is nonconvex and trying all possible solutions is computationally costly even for standard-sized problems; hence, it is declared an NP-hard problem. However, other approaches have been suggested in the literature, under some conditions on the measurement matrix Φ, relaxing the optimization in (2.9) to the convex ℓ_1-minimization approach, as shown in (2.9). Solving ℓ_1-minimization problems in near polynomial time can be achieved using linear programming solvers, and the ℓ_1-norm approach is sparse, while the ℓ_2-norm minimum approach is not [70]. The CS reconstruction algorithm returns the sparse representation of the signal of interest x, such as $\hat{s}$, and the restored signal $\hat{x}$ can be achieved from $\hat{s}$ by applying its inverse transform.

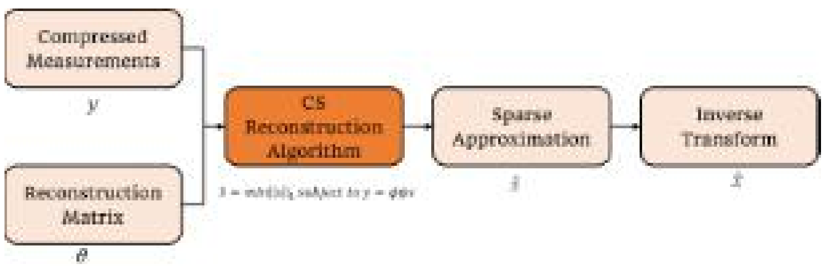

Figure 2.8: CS Reconstruction Model

In some well-known basis or dictionaries, the CS reconstruction algorithm is used to determine the sparse estimation of the original input signal from the compressed measurements. Researchers have been studying this aspect of CS to develop a better algorithm to achieve a sparse solution. Several state-of-the-art CS algorithms are presented, such as OMP [38], IHT [71], GP [72], and BCS [73]. There are several factors driving research in this area. These include the ability to reconstruct from a minimal number of measurements and the robustness of the system against noise, speed, complexity, and performance requirements [74–76]. The reconstruction algorithms can be categorized into six types, as shown in figure 2.9. A significant aspect of the CS framework is effectively recovering the original signal from compressed measurements. Convex class algorithms solve convex optimization problems through linear programming. Exact reconstruction requires a small number of measurements, but the methods are computationally complex. Nonconvex problems are commonly encountered in practical situations, and most nonconvex problems are difficult to solve within a reasonable time. In recent years, iterative greedy algorithms have been widely used in compression sensing because of their fast reconstruction and low complexity. A combinatorial or sublinear algorithm recovers sparse signals by group testing. Compared to convex relaxation and greedy algorithms, these algorithms are extremely fast and efficient, but they require a specific pattern in the measurements, which should be sparse. CS recovery problems can be solved more rapidly by iterative approaches than convex optimization techniques. Soft or hard thresholding is used to recover correct measurements from noisy ones if the signal is sparse. Thresholding depends on the number of iterations and the nature of the problem. For instance, Bregman's method is an effective iterative algorithm for solving convex optimization problems. Using these algorithms, the problem of l1 minimization can be solved efficiently and straightforwardly. In CS framework, the measurement of the original signal is achieved by multiplying the signal with a predefined measurement matrix. Reconstruction from these measurements to the original signal involves solving an underdetermined system, particularly when the signal is sparse in certain domains. Key aspects of traditional CS recovery methods include: Enhancing Signal Sparsity: Traditional CS methods require signals to be inherently sparse or to become sparse after a specific transformation. Identifying an appropriate sparse representation method is crucial, especially since not all signals are naturally sparse. Common methods for achieving sparsity include wavelet transform [77], Fourier transform [78],Gabor dictionary [79], Curvelet transform [80],short-time Fourier transform [78], Beamlet transform [81], Contourlet transform [82], K-Singular Value Decomposition (K-SVD) algorithm [83]

Designing the Measurement Matrix: It is essential to design a measurement matrix that is hardware-efficient and satisfies the Restricted Isometry Property (RIP). This ensures accurate signal reconstruction.

Solving the Optimization Problem: CS reconstruction approaches often convert the non-convex optimization problem into a convex one by modifying the objective function or relaxing some constraints, making it easier to solve.

While such CS methods effectively reduce data redundancy and energy costs on sensing side but they face practical challenges on reconstruction side. The measurement matrix may lack structural information about the original signal, leading to suboptimal reconstruction performance, especially at low sensing rates. Additionally, the storage requirements for the measurement matrix, computational power, and reconstruction time pose significant challenges.

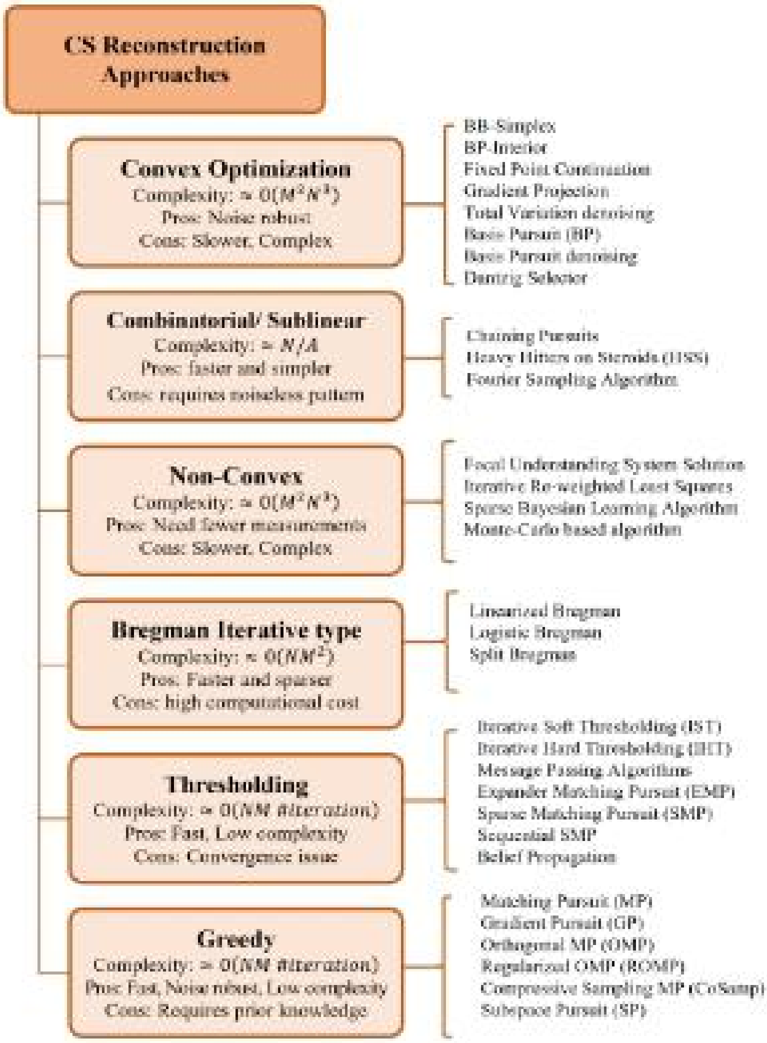

Figure 2.9: Reconstruction Algorithms

Deep Compressive Sensing

With the advent of deep learning, new approaches for CS signal recovery have emerged. In deep learning models, such as Convolutional Neural Networks (CNNs), the process is entirely data-driven and highly adaptive. Unlike traditional CS, the original signals do not need to be sparse in nature or in a transformed space. These signals can be natural images, medical signals, and more. Convolutional layers or fully connected layers replace the measurement matrix, with layer parameters learned adaptively. The reconstruction process is transformed into an end-to-end mapping, which is learned automatically during network training.

Deep learning-based methods often achieve higher reconstruction accuracy compared to traditional CS methods by leveraging the relationships between data points. The end-to-end training approach eliminates the dependency on sparse representation, simplifying the overall process. However, reconstruction performance at low sensing rates remains a challenge. To address this, we propose a novel model designed to enhance reconstruction quality discussed in chapter 5.

Compressed Learning

Compressed Learning (CL) integrates compressed sensing (CS) and machine learning to enable classification directly within the measurement domain, thereby avoiding the computational cost of reconstructing data to its original high-dimensional form [84]. Theoretically, CL can reduce the computational load of complex neural networks, making it well-suited for on-device classification. Unlike traditional CS-based sensing systems, CL not only uses CS to generate compressed data and reduce the energy required for input dimension reduction in classification tasks, but also eliminates the reconstruction stage, further cutting down on transmission energy. It is noteworthy that the application of CL in on-device classification has been recently proposed and is currently limited to simple functions such as electrocardiogram (ECG) abnormality monitoring [85]. Various other studies [86–88] have been conducted to analyze physiological signals directly on CS measurements without the need for time-consuming signal reconstruction. Essentially, CS reconstruction involves learning the signal from linear CS measurements using iterative algorithms, and integrating this learning with ML can be a clever approach. However, there has not yet been a systematic approach to compressed learning for direct features extraction from the physiological signals, although some studies have implemented compressed learning for identification purpose . Ching-Yao Chou et al. [89] proposed a biometric user identification method using ECG, known as the Compressed Alignment-Aided Compressive Analysis (CA-CA) algorithm. Therefore, developing a CL framework with energy-efficient hardware capable of supporting complex neural networks remains an area for future research. We discussed the CL in Chapter 6.

2.7 Conclusion

This chapter highlighted the principles and advancements of compressive sensing, particularly its role in biomedical signal processing with a focus on electrocardiogram (ECG) signals. CS was presented as a groundbreaking approach that challenges conventional Nyquist-based sampling by enabling simultaneous signal acquisition and compression, addressing the inefficiencies of traditional data handling methods.

The foundational aspects of CS were explored in detail, emphasizing the reliance on signal sparsity, the careful selection of sensing matrices, and the implementation of efficient reconstruction algorithms. Concepts such as the Restricted Isometry Property (RIP) and the incoherence of measurement matrices were identified as pivotal for ensuring successful and accurate signal reconstruction. Additionally, structured matrices like Fourier and Hadamard matrices were highlighted for their practical advantages, including ease of reproducibility and reduced computational overhead.

The application of CS to ECG signal processing was analyzed, showcasing its ability to significantly reduce data transmission and storage requirements without compromising diagnostic accuracy. The CS framework comprising encoding, storage, reconstruction, and feature extraction were systematically detailed, demonstrating its adaptability to real-world healthcare scenarios.

Emerging paradigms such as Deep Compressive Sensing and Compressed Learning (CL) were also discussed. These approaches extend the capabilities of CS by integrating machine learning techniques to enhance reconstruction accuracy and streamline feature extraction directly from compressed data. This innovation eliminates traditional reconstruction bottlenecks, offering scalable and energy-efficient solutions ideal for real-time healthcare monitoring systems.

In conclusion, this chapter has laid a strong theoretical and practical foundation for the use of compressive sensing in biomedical applications. The insights gained from this discussion set the stage for further exploration into advanced techniques and their integration into robust, energy-efficient, and scalable healthcare technologies.

Chapter 3

Compressive Sensing for Secure and Energy Efficient ECG Signals

3.1 Introduction

Internet of Medical Things (IoMT) has revolutionized the field of health care, introducing new technologies that can support doctors in better manage patients' diseases by continuously monitoring their vital signs. Among the observable parameters, Electrocardiogram (ECG) signals provide invaluable insights into the electrical activities of the heart, playing a pivotal role in diagnosing various cardiac anomalies, including arrhythmia. In recent years, Wireless Body Sensor Networks (WBSNs) for ECG monitoring have gained significant popularity [90–93]. This technology allows connecting wearable sensor nodes, able to sample and transmit the physiological signals, to a remote system responsible for real-time monitoring and diagnostics. Traditional methods for ECG signal acquisition and processing most often involve a quite large amount of data that require significant storage and transmission resources. This poses severe challenges, especially in real-time monitoring and telemedicine applications where efficiency on this side is paramount in view of reducing energy consumption. Moreover, given the sensitivity of the data coming from sensors, security during the transmission phase is another issue to be faced. CS has emerged as a promising paradigm to address both these challenges. By leveraging the inherent sparsity of physiological signals, CS facilitates the acquisition of signals at a rate much lower than the Nyquist rate, without compromising the ability to accurately reconstruct the original signal [9]. Thanks to its ability of compressing the information during the signal acquisition phase, CS allows not only to reduce the data volume but also to streamline the subsequent processing tasks [94, 95]. In the perspective of WBSNs, where miniaturized hardware devices are responsible to locally collect data and then transmit information to the processing unit, CS allows thereby to reduce the required computational effort, thus extending the node lifetime because of the improved energy efficiency [96]. Moreover, because of its theory based on the projection of acquired samples into a random subspace, the CS technique has also demonstrated valuable security capabilities [97, 98].

In this chapter, we present a secure and energy-efficient system relying on the CS paradigm applied to ECG signals. With the objective to reduce data volume on the side of the energy-constrained wearable sensor node, while ensuring secure transmission and accurate signal reconstruction and diagnostics on the side of the remote system, this research work provides the following contributions

- A comprehensive analysis of the CS design space, including more than 550 possible combinations between matrices used for the compression, reconstruction algorithms and sparse basis, by which we highlight the trade-offs between reconstruction quality and rate of compression achievable through CS
- A detailed evaluation of the sensing matrices, reconstruction algorithms and sparse basis on whole MIT-BIH Arrhythmia Database. Obtained results show good performances compared to state-of-the-art CS- and non-CS-based techniques
- A simple yet efficient Bi-level encryption method to enhance the level of security during the transmission
- Low-cost hardware implementations of the proposed crypto-system based on the ST Nucleo F302R8 microcontroller and the Xilinx XC7A12T FPGA chip. In comparison to similar hardware implementations, the proposed design achieves very competitive quality-compression trade-offs, while exhibiting remarkable energy performances.

3.2 Related works and Motivation

In the past years, a significant amount of research works has focused on the CS theory applied to ECG signals. The study by Fira and Goras [99] explored methods for acquiring ECG signals, using projection matrices, and reconstruction dictionaries. They proposed two approaches: one that captures the ECG signal directly without preprocessing (called patient-specific classical compressed sensing), and another that preprocesses the signal by dividing it into individual heartbeats to improve sparsity (called cardiac patterns compressed sensing). The latter was identified as the best method, while, in general, it was found that using specific dictionaries containing ECG segments allows to significantly enhance the reconstruction quality. The research work [100] used various wavelet families, particularly emphasizing the reverse biorthogonal wavelets. Using the Random Gaussian Matrix for compression and the L1-norm minimization technique for recovery, this research found the *rbio.3.7* wavelet to be the most effective. Similarly, in [101], authors employ multiscale CS for ECG signals, using a random sensing matrix with Gaussian distribution entries and wavelet domain as the sparse basis. Reconstruction is achieved through L1-norm minimization, and evaluations indicate that the method, guided by multiscale entropy, effectively captures ECG signal information. In [102], a multiscale CS (MSCS) method for multichannel ECG signals, utilizing eigenspace as the sparsifying basis, especially for high-frequency subband signals has been demonstrated. The reconstruction employs the orthogonal matching pursuit (OMP) algorithm. The proposed method enhances compression performance, retains clinical features in the reconstructed signal, and offers flexibility in data rate control, achieving satisfactory results with minimal distortion. Moreover, thanks to its reduced computational complexity, the OMP algorithm represents a competitive alternative to the L1-norm minimization one. The paper [103] employs Wavelet Transform (WT) combined with a CS matrix to encode and compress ECG signals in wireless environments, and reconstructs them using the basis pursuit denoise algorithm. Findings indicate that wavelet basis selection is crucial, with *rbio4.4* outperforming others at high compression rates. The study suggests that averaging PRD is not suitable for evaluating ECG mHealth solutions due to high variance in quality results. Picariello *et al.* [104] propose a frame-adaptive sensing matrix, with the form of a circulant matrix, in order to guarantee that the significant portions of the signal waveform are actually contained in the compressed version. In [105], the adaptive capability of the sensing matrix was implemented through a feedback scheme: during the normal operating process, the receiver is responsible to detect channel quality and possible congestion and to send such an information to the compression side; the latter can change at runtime the adopted sensing matrix, choosing between at least two possible modes, in order to relax/stress the reconstruction quality constraints depending on the current status of the network.

Numerous researchers have utilized various DWT base functions for ECG signals [54], [106–115]. Many authors have applied DWT in CS for other applications [116–118]. Some have also posited that ECG signals exhibit sparsity in the time domain [119]. The studies proves that CS method reduces computational requirements in portable settings. Thus, integrating CS with DWT for portable ECG systems is beneficial, leveraging the sparsity of the ECG signal in the wavelet domain and employing a compression method with minimal computational overhead. Of course, the research in this field universally agree that the choice of sensing matrices, sparse basis and reconstruction algorithms plays a crucial role in determining the efficiency of the CS process and the quality of the reconstructed signals. Different matrices offer various levels of incoherence with the orthogonal basis, which in turn affects the reconstruction accuracy. Similarly, the efficacy of reconstruction algorithms is contingent upon their ability to exploit the sparsity of the signals in a given domain.

Several works [106, 120–122] have also focused on low-cost hardware implementations of CS, including microcontrollers and FPGAs as target technology. The proposed systems rely on both methodological (i.e. choice of the sensing matrix, sparse basis and reconstruction algorithm) and architectural (i.e. data quantization, skipping of redundant operations, etc.) optimizations, with the objective of improving either speed or energy performances or both. However, in most of the cases, such design choices are paid with a lowering of the quality score (QS), which establishes the ratio between the compression factor and the reconstruction quality.

Finally, a very limited number of prior researches utilizes CS to enhance the security level [122, 123]. In [122], authors observed that using a fixed sensing matrix for the entire transmission process may result in poor secrecy, since an attacker could run an intensive search of all matrices that decipher the signal in order to identify the matching one. In order to increase the randomness of the sensing matrix, the usage of a linear shift feedback registers was proposed [122]. Similarly, the Wireless Body Area Networks (WBANs) scheme discussed in [123] enhances security through a novel integration of semitensor CS, chaotic scrambling, and Arnold scrambling, significantly reducing data storage and transmission requirements while improving encryption. However, both such designs rely on quite complex extra operations, which in turn could undermine the energy saving coming from CS.

The above analysis highlights that, although CS is a well-established approach and its principle is applicable in any context dealing with sparse signals in the wavelet transformed domain (biomedical signals, cognitive radio, imaging, etc.), there are still several aspects worthy of further investigation. First, for the target signal type, that in our study is the ECG, selecting an appropriate combination of wavelet base, sensing matrix and reconstruction algorithm, suitable to satisfy both the reconstruction quality and the hardware performance of the target application, is still a challenge. Second, a simple yet efficient solution to enhance the encryption level of the signal during its transmission is still missing. Therefore, in this research, we delve deep into the application of CS for the acquisition and reconstruction of arrhythmia ECG signals. We evaluate a diverse set of sensing matrices, namely Random Gaussian Matrix (RGM), Random Sparse Binary Matrix (RSBM), Random Bernoulli Matrix with signed indices (RBMS), and Low Density Parity Check (LDPC) Matrix. Alongside, we also assess the performance of the reconstruction algorithms, including Convex optimization using L1 (CVXL1), Orthogonal Matching Pursuit (OMP) and Iterative Hard Thresholding (IHT). Furthermore, given the significance of the sparse basis in CS, we explore a comprehensive set of 46 wavelet orthogonal bases. These include the haar, Daubechies (*dbn*, where n ranges from 2 to 10), symlet (*symn*, with n ranging from 2 to 8), coiflet (*coifn*, with n ranging from 1 to 5), Biorthogonal (*biornr.nd* with varying values of nr and nd), and reversebior (with specific combinations of nr and nd). For the best approach identified by this analysis, we also present energy-efficient hardware implementations based on low-cost microcontroller and FPGA devices. Through this research, we aim to shed light on the optimal combinations of sensing matrices, reconstruction algorithms, and wavelet bases in order to achieve highly accurate and hardware efficient acquisition and reconstruction of arrhythmia ECG signals using CS. Additionally, we proposed CS-Based Cryptosystem to enhance the security and ensure the ECG signal privacy and integrity during transmission.

3.3 Methods and Materials

3.3.1 Dataset

ECG signals available through the well-known MIT-BIH Arrhythmia Database have been used as a benchmark dataset [124, 125]. It containing 48 half-hour excerpts of two-channel ambulatory ECG recordings, sampled at a Nyquist frequency of 360Hz. This database is widely recognized as very effective for its comprehensive collection of arrhythmia signals, which are crucial for the development and evaluation of signal processing techniques. The effectiveness of CS algorithms is significantly influenced by the sparsity level of the signal. However, the latter depends on the number of samples are included in each of the segments in which the input signal is partitioned. Therefore, we divided the records into segments with segment length 1024 samples.

3.3.2 Compression stage

In our design space the following four distinct sensing matrices have been considered. Their impact on the reconstruction quality has been evaluated for each of 46 wavelet bases mentioned in Section 3.2.

Random Gaussian Matrix (RGM)

A Random Gaussian Matrix is defined where each entry Φ_{ij} is drawn from a normal distribution:

$$\Phi_{ij} \sim \mathcal{N}(0, \frac{1}{M}) \tag{3.1}$$

where M is the number of measurements.

Sparse Binary Matrix

The sensing matrix Φ is generated as a random binary matrix using Bernoulli trials. Each entry Φ_{ij} is determined as follows:

$$\Phi_{ij} = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases} \tag{3.2}$$

where $p = 0.1$ represents the probability of a '1' in the matrix. The generated matrix Φ has dimensions $M \times N$, where M is the number of measurements and N is the length of the signal.

Random Bernoulli Matrix with (± 1) Indices

In order to achieve signed entries, the above definition is replaced by:

$$\Phi_{ij} = \begin{cases} 1 & \text{with probability } p, \\ -1 & \text{with probability } 1-p. \end{cases} \tag{3.3}$$

There, the sensing matrix Φ is generated with entries drawn from a Bernoulli distribution and then signed, where p is the probability of an entry being 1 before signing.

LDPC Matrix

A column regular LDPC sensing matrix Φ is a binary matrix holding a constant number of 1's in each column and significantly lower than the matrix dimension itself. In our implementations such a number is set to 4. This ensures that the matrix Φ has uniform sparsity, which is a key property for LDPC codes used in error correction and compressive sensing.

3.3.3 Reconstruction stage

After compressing the signal with before mentioned sensing matrices, we used two reconstruction algorithms such Convex Optimization with $L_1 - norm$ (CVX-L1) defined in Algorithm 1 and Orthogonal Matching Pursuit (OMP) defined in Algorithm 2. Further we explores the sparsity of in 46 different wavelet basis. Additionally, we also used Block Sparse Bayesian Learning (BSBL) algorithm for the reconstruction of signals that demonstrate block-sparse structures, characterized by clusters of non-zero coefficients. Unlike CVX and OMP sparse recovery methods, which typically depend on a specific transform basis, to detect sparsity, BSBL operates independently of any designated domain. Instead, it utilize the intrinsic block-sparse properties of the signal directly within its original domain. This approach allows BSBL to adapt flexibly to the unique structure of the data, making it particularly effective for applications where sparsity is naturally organized in clustered patterns.

Algorithm 1 Convex L1-Minimization

Input: Compressed measurements $y \in \mathbb{R}^M$, Measurement matrix $\Theta \in \mathbb{R}^{M \times N}$
Output: Reconstructed signal $x \in \mathbb{R}^N$
Initialize: $x^0 \in \mathbb{R}^N$, tolerance $\epsilon > 0$, maximum iterations $N_{\max}$
for $k = 0$ to $N_{\max} - 1$ **do**
    Solve the L1 minimization problem:
        $x^{k+1} = \arg\min_x \|x\|_1$
        subject to $\|\Theta x - y\|_2 \leq \epsilon$
    if $\|x^{k+1} - x^k\|_2 < \epsilon$ **then**
        break
    end if
end for
return x^{k+1}

Algorithm 2 Orthogonal Matching Pursuit (OMP)

Input: Compressed measurements $y \in \mathbb{R}^m$, Sensing matrix $\theta \in \mathbb{R}^{m \times n}$, Sparsity level s
Output: Reconstructed signal $\hat{x} \in \mathbb{R}^n$
Initialize the residual $r^0 \leftarrow y$, the support set $\Lambda^0 \leftarrow \emptyset$, solution $\hat{x}^0 \leftarrow 0$, and iteration counter $k \leftarrow 0$
while $k < s$ **do**
    Find the index j that maximizes correlation with the residual:
        $j \leftarrow \arg\max_i |\langle \phi_i, r^k \rangle|$
    Update the support set:
        $\Lambda^{k+1} \leftarrow \Lambda^k \cup \{j\}$
    Solve the least squares problem:
        $\hat{x}^{k+1} \leftarrow \arg\min_z \|\theta_{\Lambda^{k+1}} z - y\|_2$
    Update the residual:
        $r^{k+1} \leftarrow y - \theta_{\Lambda^{k+1}} \hat{x}^{k+1}$
    Update iteration counter:
        $k \leftarrow k + 1$
end while
Reconstruct the full signal $\hat{x}$ by assigning $\hat{x}^{k+1}$ to the indices in Λ^{k+1} and zeros elsewhere
return $\hat{x}$

3.3.4 Quality Evaluation

To assess the performances of the various solutions a set of commonly used metrics have been adopted. To this purpose, we evaluated the quality of reconstruction as a function of compression capability expressed as the Compression Ratio (CR). The CR is defined as the proportion of the number of measurements M obtained through the compression process to the number of original signal data points N, reflecting the extent of data size reduction:

$$CR = \left(\frac{N - M}{N}\right) \times 100\% \tag{3.4}$$

The quality of the reconstructed signal has been measured through the Percent Root-mean-square Difference (PRD) and the associated Signal-to-Noise Ratio (SNR) that quantify the percentage of error between the original signal vector x and the reconstructed signal vector $\hat{x}$. They are calculate as shown in (3.5) and (3.6), respectively.

$$PRD = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \times 100 \tag{3.5}$$

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2} \tag{3.6}$$

Such metrics are known as "non-diagnostic error measures" and they are very commonly used in evaluating the signal quality for medical use [104,105,126–129]. Previous works have demonstrated that they effectively predicts the subjective quality rating given by the cardiologists. Through semiblind tests in which cardiologists evaluate reconstructed signals from different compression algorithms by comparing each of them to the original ones, appropriate thresholds to evaluate the quality of the reconstructed signals were proven, and largely approved in the current literature [130]. Further information can be obtained through the "diagnostic error measures", such as the Weighted Diagnostic Distortion (WDD) and the Wavelet Energy Diagnostic Distortion (WEDD), as defined in [130] and [131], respectively. They consider the diagnostic distortion of the local waves such as P-wave, Q-wave, QRS complex, ST segment and T-wave of the reconstructed ECG signal and allow evaluating the preservation of important pathological features in the compressed signals. Table 3.1 summarizes thresholds levels commonly adopted to classify reconstructed signals on the basis of the PRD, WDD and WEDD measures.

Table 3.1: Diagnostic quality as a function of PRD, WDD and WEDD values.

PRD (%) [130]	**WDD (%)** [130]	**WEDD (%)**Now [131]	**Signal quality**
n.a.	n.a.	0–4.517	Excellent
0–2	0–2.3	4.517—6.914	Very good
2–9	2.3–12	6.914-–11.125	Good
9–19	12–18	11.125-–13.56	Not good
19–60	18–40	> 13.56	Bad

Finally, to assess the linear relationship between the original and reconstructed ECG signals the Correlation Coefficient (CC) as defined in (3.7) is also calculated, values close to 1 indicates a high similarity between the two signals.

$$CC = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(\hat{x}_i - \bar{\hat{x}})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{N} (\hat{x}_i - \bar{\hat{x}})^2}} \tag{3.7}$$

3.4 Proposed CS Based Cryptosystem for ECG Signals

The proposed system conjugates CS compression and secure transmission and storage of data in IoT healthcare applications by adopting a two level of encryption. Many researchers proposed a CS based encryption systems [132–134] and suggest that security level depends on the type of sensing matrix used for compression or encryption. According to essential condition of CS, reconstruction can be guaranteed if the sensing matrix is random. But, random sensing matrices leaks the energy of the original signal [135]. Therefore in this study we proposed a bi-level encryption system. In first approach, column permutation was applied to original signal with key kc followed by CS encryption with sensing matrix key Φ and in second approach CS based encryption was applied first with sensing matrix key Φ followed by column permutation on compressed signal with key kc. Figures 3.1 and 3.2 shows the simple block diagram of both approaches. There, blocks highlighted in light pink and green have been object of our design.

Figure 3.1 shows the Pre-CS encryption, it can be observed that two encryption/decryption levels are adopted. At the encoder side, Alice, the sender, initially exploits the columns permutation on the original signal x using key kc, the permuted signal pass through the additional encryption block performing CS based encryption using sensing matrix Φ key. This approach guarantees that the transmitted message cannot be easily decoded by an eavesdropper, Eve, which seeks to intercept the communication on the insecure channel. If Eve reconstruct the signals from compressed measurements that will not represent the original information as original signal x are exchanged in position based on the kc key information. Then, at the decoder stage, *Bob*, the receiver, first reconstruct the received signal based on the Φ key; then it utilizes revert column permutation with key kc.

Figure 3.2 shows the Post-CS Encryption, At the encoder side, Alice, the sender, initially exploits the CS to compress the input ECG signal x through the sensing matrix, Φ as encryption key. Then, the compressed signal passes through an additional encryption block performing columns permutation on the y vector. The latter guarantees that the transmitted message cannot be easily decoded by an eavesdropper, Eve, which seeks to intercept the communication on the insecure channel. To this aim, the elements forming the compressed signal y are exchanged in position based on the kc key information. Then, at the decoder stage, *Bob*, the receiver, first reverts the elements of the received signal based on the kc key; then it utilizes the appropriate reconstruction algorithm and the sparse base Ψ, to reconstruct the original ECG signal x.

3.5 Performance Evaluation of Pre-CS Encryption

In this approach, we used three sensing matrices for CS based compression or encryption: Random Gaussian Matrix (RGM), Random Binary Matrix with a density of 1's being 20% RBMB(d=20%), and Random Binary Matrix with a density of 1's being 10% RBMB(d=10%). We used the BSBL reconstruction algorithm for signal decoding at the receiving side. First of all, we evaluate the selection of the proper block size in the BSBL algorithm, which is most important for reconstruction. We considered block sizes ranging from 2 to 30 and measured the correlation coefficient, as shown in figure 3.4(b) the correlation

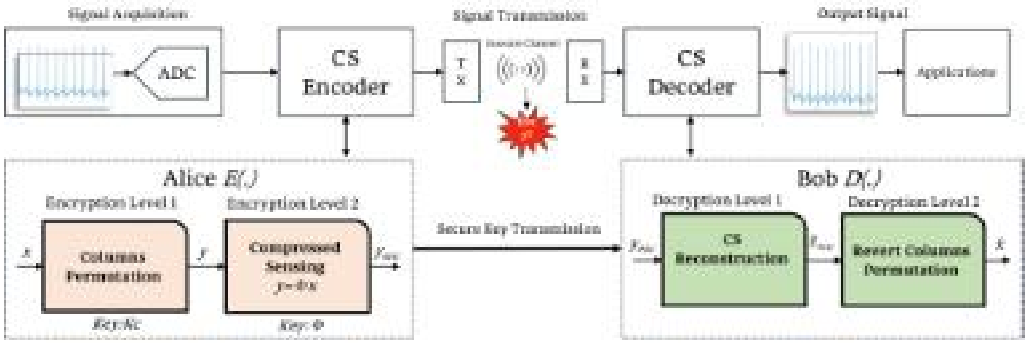

Figure 3.1: Pre-CS Encryption, Column permutation followed by CS Encryption

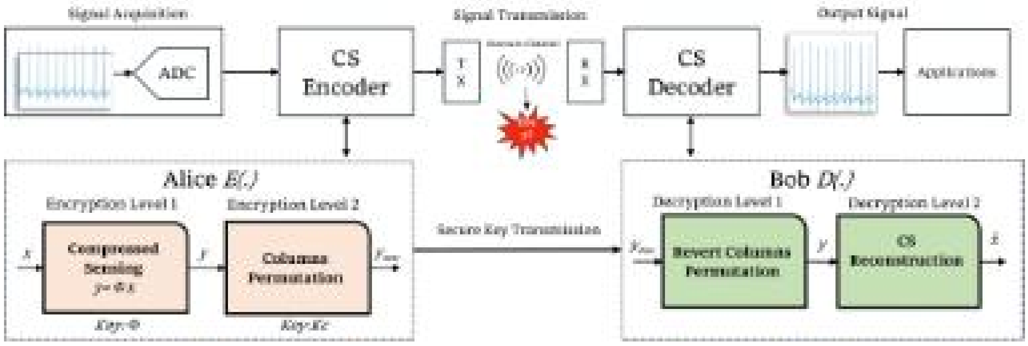

Figure 3.2: Post-CS Encryption, CS Encryption followed by Column Permutation

between the original and reconstructed signals increases quadratic with block size until it reaches 10, after which it remains approximately constant so we used block size 10 for our experiment. Later we evaluate the PRD and Correlation Coefficient between original and reconstructed signal, while the original signal was compressed with before mentioned sensing matrices with compression ratio ranging from 10% to 90%. From the figure 3.3 it shows that the PRD exponentially increasing for all matrices as the compression ratio increases. RGM consistently has the highest PRD values across all compression ratios, suggesting that its reconstructed signal differs more from the original data compared to the RBMB methods. RBM(d=20%) generally has the lowest PRD and highest CC, especially at higher compression ratios. The correlation values are stable and grater than 0.98 for all sensing matrices upto 50% compression ratio, then decrease with increase in CR ranging from 50% to 90%. The correlation indicating that its reconstructed signal is most similar to the original data among the three metrics. RGM has the lowest correlation values at higher compression ratios, while RBMB has higher CC values among all compression ratios.

The results presented are obtained from CS reconstruction as shown in green block of the figure 3.1 after reconstructing the signal we applied revert column permutation to get the original signal. For completeness, reconstructed signals at compression ratio 60% shown in figure 3.5. In this figure, the first row shows the original signal which goes through column permutation, second row shows the columns permuted signals where positions of elements are shifted. The third row shows the compressed signal from three sensing matrices and fourth shows decompressed signal with BSBL algorithm and finally last row show the decrypted signal as output of reverted column permutation block. Additionally, we introduced Additive White Gaussian Noise (AWGN) to the compressed measurements obtained at phase transition point (CR=60% where the PRD values are less than 9%), ranging from 5dB to 100dB, as shown in figure 3.4(a). The MSE decreases exponentially for all matrices with increasing SNR up to 30dB for RGM and 60dB for RBMB matrices, after which it becomes stabilized.

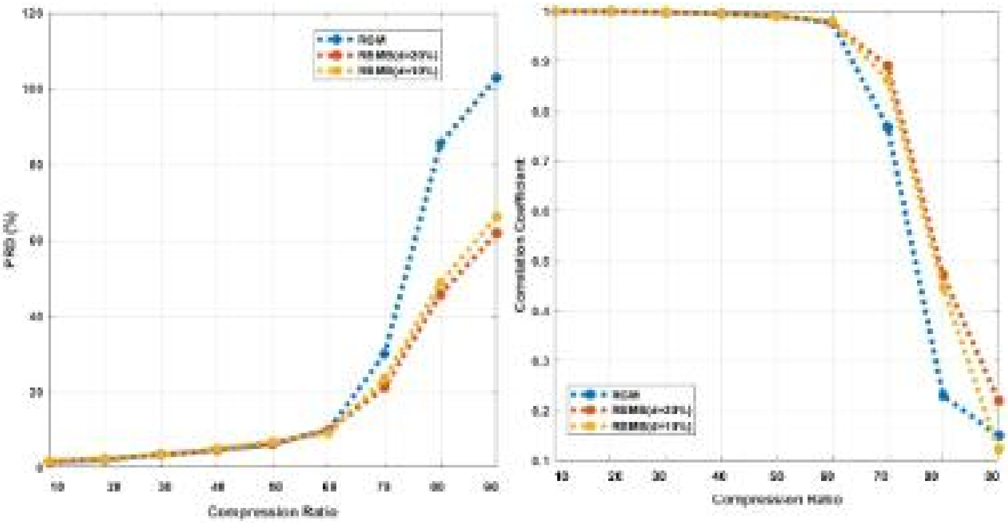

Figure 3.3: PRD and Correlation Trend at various Compression Ratios

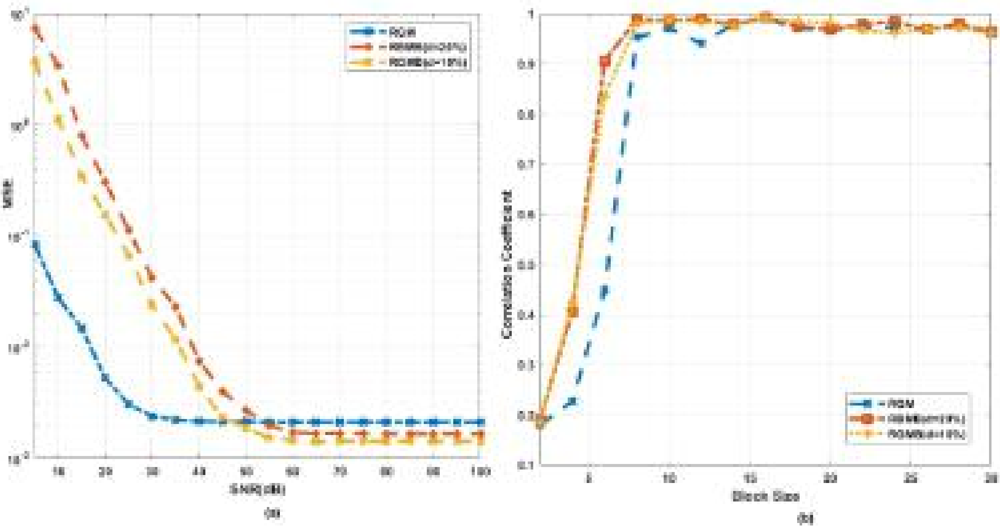

Figure 3.4: (a) Mean Square Error trend vs Adeptive White Gaussian Noise (SNR) (b) Correlation Coefficient trend vs Block Size of BSBL algorithm

3.5.1 Discussion

In the literature, authors directly used CS as encryption technique and is characterized by a single layer of encryption. Direct CS based encryption is quite simple but highly depends on the types of sensing matrix. Considering the CS framework and essential conditions of CS, it's necessary to have random

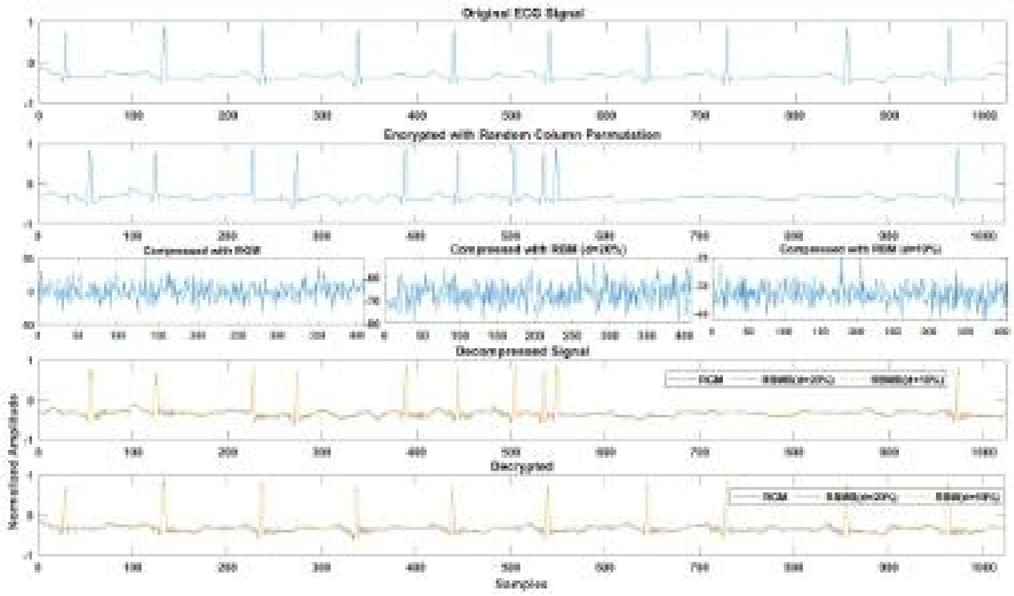

Figure 3.5: Original, Compressed and Recovered Signals from Approach 1

sensing matrix for perfect recovery but at the same time random sensing matrix leaks the energy of the original signal therefore it needs additional lightweight encryption. We proposed a dual-layer encryption mechanism, comprising both random column permutation and CS encryption which enhanced security due to dual-layer encryption. Based on the analysis of experimental results, a compression ratio (CR) of 60% yields a high-quality reconstructed signal. This implies that 40% of the data is preserved, resulting in a 60% energy savings compared to traditional methods. Furthermore, the security level is also enhanced. In just CS based encryption system if sensing matrix Φ is Binary then key would be $M \times N$ bits, but if it's not binary then it requires $(M \times N) \times b$ bit key where b is the bit quantization level. This key can be transmitted once over a secure channel. It's challenging to decipher the key since retrieval requires might be possible in $M \times N \times b$ different attempts. In contrast, our proposed Pre-CS encryption introduces an additional encryption layer through column permutation with key k_c with size N. Even if one manages to retrieve the compressed signals in $M \times N \times b$ number of attempts, the information within the signal will be hidden. It would then require decryption using key k_c. A limitation of Pre-CS encryption is that it can only be applied to preprocessed signal. In the real time signal need to be acquired first then shift the positions based on column permutation key and then CS technique can be applied. Furthermore, Pre-CS encryption is beneficial for storing or transmitting the Pre-processed data in compressed and encrypted form to save the hardware resources and channel bandwidth. To apply the approach in real time continuous monitoring we proposed Post-CS encryption method which is described in the next section.

3.5.2 Performance Evaluation of Post-CS Encryption

For Post-CS Encryption we used four sensing matrices such as RGM, RBMB, RBMS and LDPC sensing matrices and two reconstruction algorithms CVX-L1 and OMP algorithm and explored the sparsity in 46 wavelet basis. In this section we discuss the analysis of our complex design space by considering CR ranging from 10% to 70%. For this ablation analysis, we employed a sub-portion of sample records from the MIT-BIH dataset and run all experimental procedures on MATLAB 2023b. Our goal is to identify the solutions providing the highest Quality Score (QS), as defined in (3.8), while maintaining the PRD at least below the 9% threshold corresponding to a "Good" reconstructed signal quality.

$$QS = \frac{CR}{PRD} \tag{3.8}$$

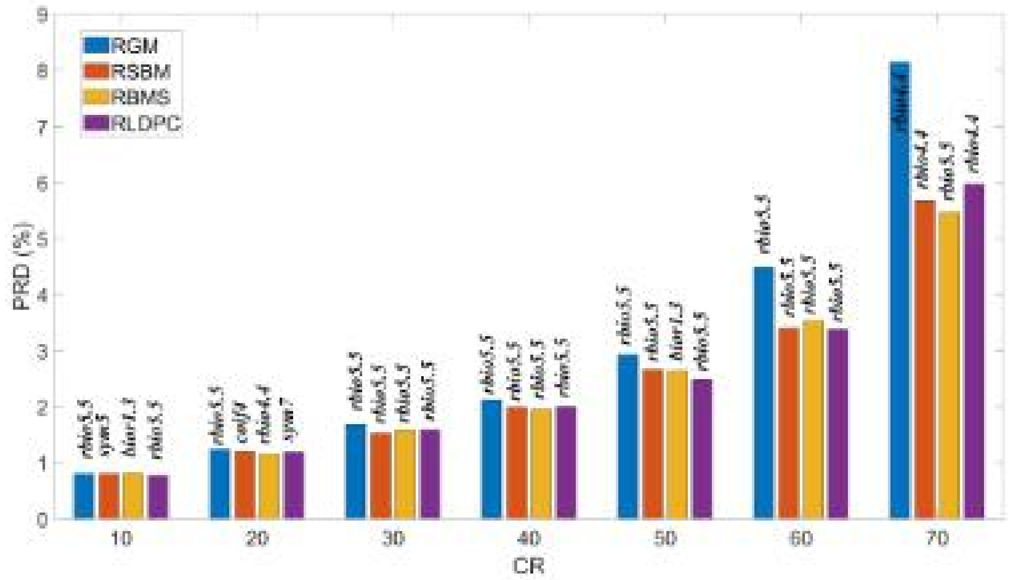

Figure 3.6: Ablation analysis comparing CVXL1 reconstruction algorithm

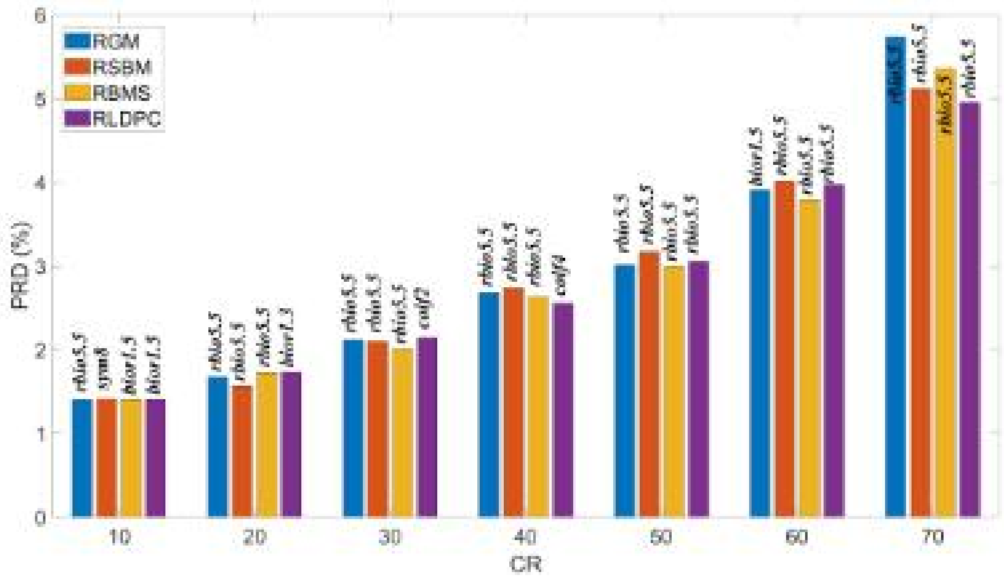

Figure 3.7: Ablation analysis comparing OMP reconstruction algorithm

Overall, figures 3.6& 3.7 provides two key information. First, focusing on the sensing matrix, the LDPC one, most often, performs better than counterparts, also at higher CRs. This is a quite nice finding, since, thanks to its structure, such a binary matrix allows to save a considerable amount of computational resources with respect to RGM and RBMS. Second, focusing on the reconstruction algorithm, CVXL1 seems to be the best option for CR up to 60%. After that, OMP overcomes its competitors by exhibiting a PRD 17% lower than CVXL1 at CR 70%. To provide the reader with some numerical results, Table 3.2 reports the QSs achieved by the LDPC-based implementations at CR=50%, 60% and 70%, considering all the two reconstruction algorithms.

Table 3.2: Best QS exhibited by the LDPC-based implementations.

	CVXL1	OMP
CR=50%	20	16.33
CR=60%	14.8	15.11
CR=70%	8.37	14.14

From the above analysis it can be concluded that, while the CVXL1 is the most efficient reconstruction algorithm at CR=50%, one could be willing to trade some signal quality, still keeping the PRD below the 9% threshold, and exploit a more aggressive compression, i.e. 70%. In such a case, the OMP algorithm, in conjunction with the LDPC matrix at the compression side, is preferred.

3.5.3 Quality evaluation for the best approaches

Now that the best candidate approaches have been identified, their detailed characterization is presented. Such results have been obtained by elaborating through the proposed system all the 30 minutes long 48 records constituting the MIT-BIH database. Table 3.3 lists the performances in terms of PRD, SNR, MSE and CC achieved for each record by using the LDPC as sensing matrix at both CR=50% and CR=70% and by using the CVXL1 and OMP reconstruction, respectively. On average, the PRD exhibited in both cases is well below the 9% threshold that, as shown in Table 3.1, defines the "Good" quality of the reconstructed signals. In figure 3.8 we report plots of original and reconstructed ECG signals belonging to 5 different records obtained using CR=50% and the CVXL1 algorithm. Among them, the critical record 212, which leads to the worst performance according to Table 3.3, is also illustrated. Visual inspection shows that also in such case all relevant signal features are well maintained after the proposed compression. Figure 3.9 and figure 3.10 show the worst cases identified in Table 3.3 for the CR=70% + OMP (Record 212) and for the CR=50% + CVXL1 by feeding the system with a signal sub-sampled at 180Hz (Record 215). While imperfections in the reconstructed signals are quite evident, it can be also observed that main characteristics of the original ECG are still maintained.

3.5.4 Comparasion with State-of-the-Art

As a final remark, Table 3.4 compares the characteristics exhibited by the proposed CS techniques with some relevant state-of-the-art competitors. The latter rely on both CS- [104, 105], DWT- [126, 127] and deep learning-based [128] ECG compression techniques. Thanks to the implementation of the column shifting method, only the proposed approach among the competitors implements a cryptosystem layer through the key kc. In terms of compression performance,

Table 3.3: Performances achieved by the proposed best approaches over all 48 records of 30-min sampled data from the MIT-BIH database. Results obtained for sub-sampled input signals at 180Hz are also reported.

	CVXL1, CR=50% (360Hz)				**OMP, CR=70% (360Hz)**				**CVXL1, CR=50% (180Hz)**			
Record	PRD (%)	CC	MSE ($\times 10^{-3}$)	SNR (dB)	PRD (%)	CC	MSE ($\times 10^{-3}$)	SNR (dB)	PRD (%)	CC	MSE ($\times 10^{-3}$)	SNR (dB)
100	6.760	0.999	0.099	24.279	6.772	0.991	0.612	23.580	4.735	0.995	0.328	26.576
101	2.976	0.999	0.151	31.054	6.307	0.994	0.706	24.666	7.957	0.991	0.934	22.788
102	3.790	0.998	0.136	28.563	9.262	0.988	0.824	20.874	6.618	0.993	0.432	23.649
103	2.690	0.999	0.111	31.520	5.883	0.997	0.564	24.832	6.155	0.997	0.585	24.352
104	4.687	0.996	0.597	28.163	11.481	0.980	3.753	20.540	7.851	0.995	0.818	22.565
105	4.010	0.999	0.364	28.258	9.822	0.992	2.392	20.652	7.860	0.995	0.907	22.343
106	3.783	0.999	0.271	28.956	8.488	0.994	1.482	22.175	7.483	0.996	0.948	22.835
107	2.500	1.000	0.510	32.397	7.123	0.997	4.239	23.381	7.149	0.997	3.723	23.030
108	5.695	0.994	0.362	26.071	12.586	0.973	1.809	19.197	8.044	0.987	0.645	22.130
109	2.393	1.000	0.172	32.754	6.295	0.997	1.230	24.457	6.420	0.997	1.028	23.949
111	4.192	0.998	0.131	27.988	9.501	0.991	0.706	21.011	7.204	0.996	0.337	22.971
112	1.541	0.998	0.174	36.496	3.743	0.986	1.082	28.926	2.887	0.991	0.632	30.866
113	2.565	1.000	0.124	31.916	5.272	0.998	0.564	25.848	4.779	0.999	0.503	26.563
114	5.628	0.996	0.138	25.510	12.311	0.980	0.699	18.767	6.798	0.992	0.192	23.553
115	2.260	0.999	0.168	33.178	5.514	0.995	1.044	25.522	3.963	0.997	0.645	28.186
116	1.837	0.999	0.407	34.960	4.633	0.995	2.641	26.986	4.522	0.997	2.553	26.946
117	1.555	0.998	0.182	36.280	3.389	0.989	0.881	29.534	2.607	0.995	0.469	31.688
118	2.743	0.998	0.790	31.926	7.618	0.983	5.902	22.959	4.772	0.992	2.647	26.502
119	1.642	0.999	0.270	35.797	4.043	0.996	1.667	28.084	3.298	0.998	1.132	29.737
121	1.410	0.998	0.127	37.455	3.135	0.991	0.645	30.584	2.066	0.997	0.295	33.791
122	1.934	0.999	0.315	34.344	4.376	0.994	1.616	27.260	3.758	0.995	0.469	28.531
123	1.693	0.999	0.206	35.495	3.775	0.993	1.044	28.572	2.955	0.996	0.631	30.618
124	1.362	1.000	0.145	37.605	2.971	0.998	0.707	30.921	2.185	0.999	0.374	33.262
200	6.246	0.997	0.615	25.151	15.400	0.982	4.007	17.531	9.664	0.994	1.624	20.608
201	3.114	0.999	0.059	30.282	6.636	0.996	0.288	23.921	7.451	0.995	0.391	22.731
202	3.219	0.999	0.115	30.050	6.675	0.997	0.540	23.831	4.341	0.999	0.147	27.412
203	5.536	0.997	0.961	26.649	13.411	0.983	5.665	19.020	13.114	0.987	6.976	19.588
205	3.355	0.998	0.142	29.642	8.332	0.986	0.908	21.815	6.753	0.990	0.693	23.489
207	3.540	0.999	0.246	29.876	7.655	0.994	1.344	23.452	5.100	0.998	0.427	26.181
208	3.538	0.999	0.360	29.841	9.159	0.993	2.533	21.749	11.258	0.993	2.984	19.416
209	6.697	0.996	0.487	24.058	17.274	0.975	3.488	16.011	11.230	0.991	1.086	19.093
210	4.125	0.998	0.199	28.222	9.198	0.992	1.040	21.374	6.759	0.997	0.364	23.647
212	7.061	0.996	0.813	24.009	18.255	0.974	5.276	15.715	14.303	0.987	2.775	16.968
213	3.524	0.999	0.603	29.329	10.367	0.993	5.522	20.218	12.135	0.990	6.661	18.487
214	3.017	0.999	0.242	30.700	6.556	0.997	1.219	24.105	5.429	0.998	0.758	25.348
215	6.987	0.996	0.567	23.836	17.933	0.975	3.776	15.662	14.633	0.987	1.888	16.835
217	2.436	1.000	0.232	32.483	5.716	0.998	1.317	25.183	5.504	0.998	1.212	25.260
219	1.901	1.000	0.200	34.761	4.685	0.997	1.273	27.017	4.020	0.998	0.975	28.002
220	2.154	0.999	0.184	33.411	5.288	0.994	1.140	25.699	4.773	0.996	0.921	26.476
221	3.618	0.999	0.158	29.063	7.748	0.996	0.765	22.549	6.508	0.997	0.520	23.811
222	5.497	0.997	0.171	25.639	12.966	0.982	1.084	18.515	8.467	0.993	0.245	21.927
223	2.043	0.999	0.172	33.952	4.932	0.996	1.031	26.419	3.806	0.998	0.670	28.419
228	6.650	0.996	0.515	24.773	15.312	0.976	2.938	17.740	10.542	0.990	1.110	20.233
230	3.897	0.999	0.241	28.372	9.486	0.994	1.544	20.798	9.585	0.994	1.384	20.527
231	4.005	0.999	0.151	28.082	9.134	0.994	0.818	20.999	9.398	0.994	1.128	20.590
232	6.603	0.995	0.187	24.133	15.725	0.970	1.130	16.696	13.770	0.961	1.135	17.343
233	3.636	0.999	0.452	29.127	9.191	0.995	2.930	21.134	9.237	0.995	2.987	20.854
234	4.530	0.999	0.248	27.081	10.732	0.992	1.406	19.617	9.508	0.994	1.062	20.592
Avg	3.68	0.998	0.297	30.07	8.585	0.990	1.871	22.835	7.070	0.994	1.278	24.193

[104], [126,127] achieve acceptable PDR/SNR results while exploiting moderate CRs. At much higher CRs, i.e. 50%, the proposed CS implementation using the CVXL1 as reconstruction algorithm exhibits a PRD 65% and 52.9% lower than [104] and [127], respectively. Moreover, Table 3.4 shows that, at CR=70% and parity of reconstruction algorithm, the proposed implementation achieves a PRD reduction by 134.5 times with respect to [105]. The latter requires a different reconstruction algorithm to lower its PRD (but falling outside the "Good" range), with a consequent impact on the complexity. Finally, by considering the deep-learning based approach [128], it can be concluded that good PRD/SNR results can be achieved only through significantly more complex models compared to the CS approach.

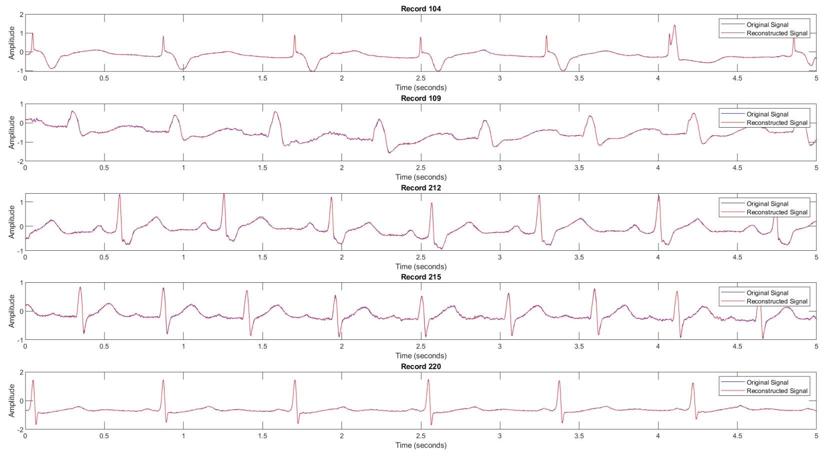

Figure 3.8: Some examples of reconstructed signals, considering the CVXL1 algorithm at CR=50%.

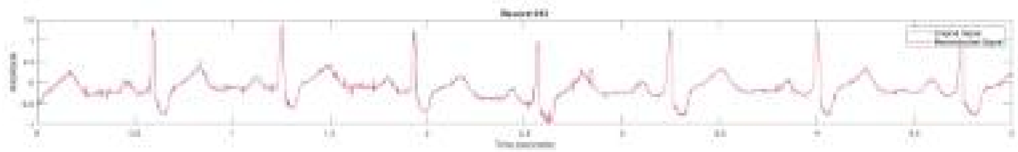

Figure 3.9: Worst case record (212) reconstructed through the OMP algorithm (CR=70%).

3.5.5 Hardware results and comparison with state-of-the-art

Two hardware implementations of the proposed CS encoder, including both the compression and the encryption operations, are here presented. The former relies on a low-cost ST Nucleo-64 development board (figure 3.11), equipped with a 64MHz maximum frequency F302R8 microcontroller unit, 64kB of Flash memory and 16kB SRAM. The software-routine describing the proposed CS encoder was written in C language. This implementation exploits the sparsity of the LDPC sensing matrix by storing only location of non-zero entries. This allows redundant computations to be skipped, thus reducing the execution time, memory requirement and energy consumption. The latter was measured by monitoring the current absorbed by the microcontroller through a Keithley DMM7510 multimeter connected to the measurement jumper IDD of the board, as shown in figure 3.11.

However, achieving real-time operation through microcontroller-based implementations may be quite difficult [106, 120, 129]. To overcome this limitation, a custom architecture has been designed for an FPGA-based implementation. The circuit implementing the proposed CS encoder was described at VHDL level and has been implemented on a low-resource Xilinx Artix 7 XC7A12T device by using the Vivado 2023.2 tools suite. Its schematic diagram is illustrated in figure 3.12. Also in this case, only the locations of non-zero entries in the sensing matrix are stored on-chip by means of the NZP memory block that, considering CR=50%, has a depth of 512. Initially, the 13-bit input data feeds the ECG samples memory buffers used to enable parallel operations along the column direction. Such a

Figure 3.10: Worst case record (215) reconstructed through the CVXL1 algorithm (CR=50%, 180Hz sampling rate).

Table 3.4: Comparison with state-of-the-art ECG compression approaches.

	Compression	Reconstruction	Complexity	CR(%)	PRD(%)	SNR(dB)
[104] (CS)	Circulant matrix	Basis Pursuit	$\mathbf{O}(N^3)$	9.91	10.62	2.98
[105] (CS)	Partial DCT, LDPC	OMP	$\mathbf{O}(kMN)$	80	6379	<-20
[105] (CS)	Partial DCT, LDPC	CoSaMP	$\mathbf{O}(MN)$	80	12.68	<-5
[126]	DWT	IDWT	n.a.	13	3	n.a.
[127]	TQWT	IDFT	$\mathbf{O}(N^2)$	30.06	7.8	n.a.
[128]	BCAE Encoder	BCAE Decoder	**11152 MACs**	117.33	7.76	15.93
This work (CS)	RLDPC	CVXL1	$\mathbf{O}(N^3)$	50	3.67	30.07
This work (CS)	RLDPC	OMP	$\mathbf{O}(kMN)$	70	8.58	22.8

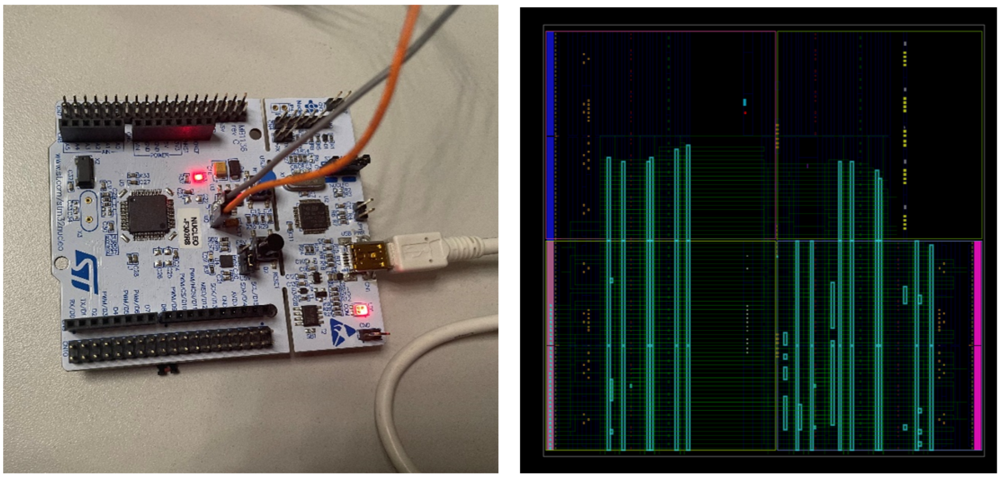

Figure 3.11: Microcontroller prototype (left) and layout of realized the FPGA chip (right).

process consumes 1024 clock cycles. During this phase, the address used to write into the ECG Samples Memory is provided by a counter that appropriately scans all the locations. After that, the CS operation starts. That is, at each clock cycle addresses provided by the NZP memory are inputted to the ECG Samples Memory blocks, which resume the corresponding signal samples; such values are then transferred in parallel to an adder tree structure that produce the compressed result. Finally, in order to implement the encryption stage, the compressed results are stored into the Support Memory, having depth 512 locations, to be properly re-organized according to the proposed column shifting method using the kc key as access address. Such a design occupies 4130 look-up-Tables, 287 flip-flops and 4 on-chip block RAMs. figure 3.13 illustrates the running of both the proposed implementations through simple timing diagrams. Considering the 360Hz sampling rate, a new sample is received every 2.77ms, resulting in an initial latency of 2.84ms before elaborations can be performed on the first segment. Due to the

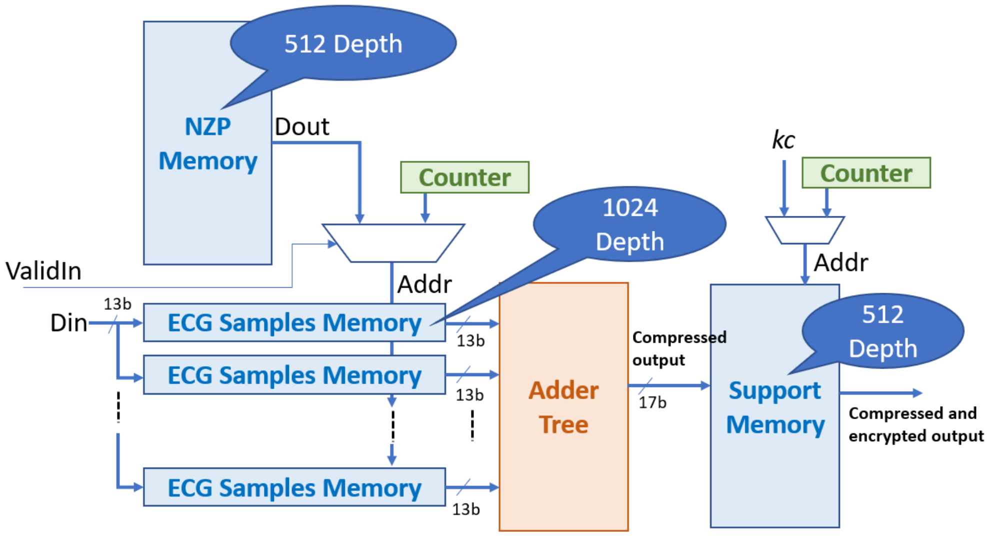

Figure 3.12: The FPGA-based circuit implementing the proposed CS encoder.

different technologies and parallelism levels, the two implementations perform quite different: the microcontroller-based one requires an elaboration time of 11.6 ms, whereas the FPGA-based one is able to accomplish all the operations within 1.03ms at 1MHz clock frequency. After the elaboration is terminated, a new segment can be transferred to each implementation. It is then clear that the microcontroller-based design is not able to sustain real-time operation without an additional five samples buffer.

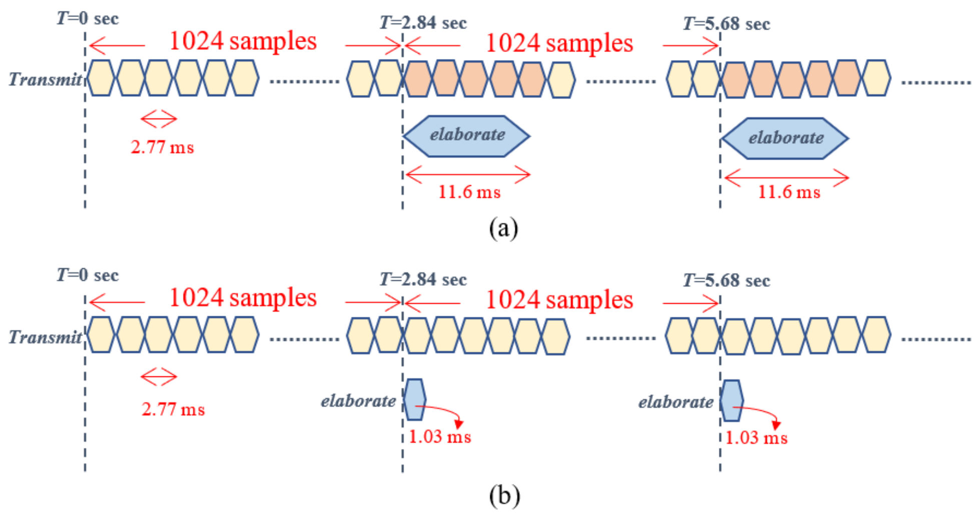

Figure 3.13: Timing diagram for the (a) microcontroller and (b) FPGA-based implementations

.

Table 3.5 collects the performances exhibited by our implementations and some relevant state-of-the-art design competitors [106, 120, 122] [129]. Overall, it can be noted that, at the parity of implementation platform, the proposed design exhibits a quite good trade-off between energy consumption, time and reconstruction quality. It is also worth noting that most of the competitors do not provide cryptosystem supports and, at the parity of CR, the proposed design shows a PRD significantly lower than the implementation [122], which relies on a linear feedback shift register to introduce the level of encryption; this gain is achieved by saving a significant amount of energy per sample.

Table 3.5: Hardware performances and comparison with state-of-the-art

	Crypto	CR (%)	PRD (%)	Platform	Time (ms)	Energy per sample (μJ)
[106]	No	71	≈ 10	uC TI MSP430	25	0.29
[120]	No	50	≈ 8	uC TI MSP430	35	312
[129]	No	3.26	≈ 0.147	uC TI MSP430	157	n.a.
[122]	Yes	50	≈ 9	uC TI MSP430	n.a.	191
				uC ST	11.6	0.97
This work	Yes	50	3.68	F302R8	1.03	0.8×10^{-3}
				FPGA Artix-7		

3.6 Conclusion

In this paper we have demonstrated the effectiveness of a CS-based cryptosystem for the secure and efficient handling of ECG signals in IoMT devices. By integrating the compressive sensing approach with a simple encryption layer we proposed a dual-purpose solution that significantly reduces the data volume for transmission and storage while ensuring the privacy and security of sensitive biomedical information. A simple encryption layer were examined before (Pre-CS Encryption) and after CS encoder (Post-CS Encoder). In Pre-CS Encryption, we explored three different sensing matrices as CS encryption key and one reconstruction algorithm allowed to identify the RBMB at 60% compression ratio provides the best solution. For Post CS-Encryption, a comprehensive comparative analysis that have explored 4 different sensing matrices, 46 wavelet bases and 2 reconstruction algorithms allowed to identify the LDPC matrix, the rbio5.5 basis and the CVXL1 reconstruction algorithm as the best solution among the whole analyzed design space at CR=50%. Indeed, in such case the proposed system achieves an average PRD=3.68 calculated on the whole MIT-BIH database formed by 48 ECG records each 30-min long. An analogous behavior has been also observed for ECG input signals sub-sampled at 180Hz. We have also proven that the approach demonstrated here can also be exploited up to CR=70% obtaining an average PRD still below the 9% threshold that corresponds to a good reconstruction quality, by adopting the OMP reconstruction algorithm at the decoder level. An FPGA prototype has been built and characterized. It was realized by using a low-resource Xilinx Artix 7 XC7A12T device. It consumes 4130 LUTs, 287 FFs, 4 BRAMs, dissipates only 0.8nJ per sample and perfectly suites the real-time operation requiring only 1.03ms per segment when running at 1MHz clock frequency.

Chapter 4

Data-Driven Sensing Matrix and Dictionary Design for Compressive Sensing of ECG Signals

4.1 Introduction

Internet of Medical Things (IoMT) has revolutionized the field of health care, introducing new technologies that can support doctors in better manage patients' diseases by continuously monitoring their vital signs. Among the observable parameters, Electrocardiogram (ECG) signals provide invaluable insights into the electrical activities of the heart, playing a pivotal role in diagnosing various cardiac anomalies, including arrhythmia. In recent years, Wireless Body Sensor Networks (WBSNs) for ECG monitoring have gained significant popularity [90–93]. This technology allows connecting wearable sensor nodes, able to sample and transmit the ECG signals, to a remote system responsible for real-time monitoring and diagnostics. Traditional methods for ECG signal acquisition and processing most often involve a quite large amount of data that require significant storage and transmission resources. This poses severe challenges, especially in real-time monitoring and such as power energy consumption, transmission data rate. Moreover, given the sensitivity of the healthcare data coming from sensors, security during the transmission phase is another issue to be faced. CS has emerged as a promising paradigm to address above challenges by leveraging the inherent sparsity of ECG signals. Thanks to its ability of compressing the information during the signal acquisition phase, CS allows not only to reduce the power consumption and data volume but also to streamline the subsequent processing tasks [95, 136]. In the perspective of WBSNs, where miniaturized hardware devices are responsible to locally collect data and then transmit information to the processing unit, CS allows thereby to reduce the required computational effort, thus extending the node lifetime [96]. Moreover, because of its theory based on the projection of acquired samples into a random subspace, the CS technique has also demonstrated valuable security capabilities [97, 98].

In CS, the effectiveness of the framework is heavily dependent on the choice of sensing matrix, the sparsity of the signal in transform domains, and the reconstruction algorithms employed. To minimize power consumption at the sensing node, selecting an optimal sensing matrix is important, while for achieving high-quality signal reconstruction, the choice of an overcomplete dictionary plays a important role. In this work, we propose a data-driven sensing matrix (DSM) and introduce two overcomplete sparse dictionaries: the Gaussian Dictionary (GD) and the K-Singular Value Decomposition (K-SVD) Dictionary. We evaluate the performance by reconstructing the compressed signal with Basis Persuit Denoising (BPDN) reconstruction algorithms. Furthermore, we convert the DSM into a Binary Thresholding Matrix (BTM) using a proposed novel thresholding approach and demonstrate its implementation on hardware. Additionally we compare the performance of with state-of-the art sensing matrices and standard dictionaries.

4.2 Related Work and Motivation

In recent years, a considerable amount of research has focused on applying CS theory to the acquisition and reconstruction of biomedical signals, particularly electrocardiogram (ECG) signals. The inherent sparsity of ECG signals in specific transform domains, such as the discrete cosine transform (DCT), Fourier, wavelet, and data-driven dictionaries, has made them an excellent sparse dictionaries for CS applications. This section provides a comprehensive review of the evolution of CS theory in relation to ECG signal processing.

Fira and Goras [99] introduced acquisition techniques based on projection matrices and reconstruction dictionaries. They proposed two approaches: one that directly acquires ECG signals (termed as patient-specific classical compressed sensing) and another that segments the ECG signal into individual heartbeats to enhance sparsity, known as cardiac patterns compressed sensing. Their results showed that utilizing specific dictionaries, particularly those containing ECG segments, significantly improves reconstruction quality. Similarly, Mishra *et al.* [100] explored the performance of various wavelet families, identifying reverse biorthogonal wavelets (*rbio3.7*) as the most effective when combined with a random gaussian matrix (RGM) for compression and L1-norm minimization for recovery. Further studies such as [101] and [102] applied multiscale CS methods using wavelet domains and eigenspace as sparse representations, coupled with orthogonal matching pursuit (OMP) for reconstruction. These works demonstrated the capacity of multiscale approaches to capture ECG signal features effectively while maintaining low computational complexity. The importance of wavelet basis selection in CS applications has been further emphasized by research from [103], which found that *rbio4.4* wavelets performed optimally at high compression rates. Additionally, D. Craven [137] proposes an adaptive dictionary reconstruction framework to enhance CS for ECG signals. By selecting appropriate dictionaries based on signal characteristics like the presence of the QRS complex, the method improves reconstruction quality and reduces power consumption compared to traditional CS techniques and SPIHT algorithms. Along with fixed sparse basis, recent studies introduced data-driven dictionaries that adapt to the statistical properties of the signal. Data-driven dictionaries, as opposed to pre-defined bases, are learned directly from the signal data and can more effectively capture unique, patient-specific features. These dictionaries are generated using machine learning techniques such as dictionary learning (e.g., K-SVD) or deep learning models, and they offer significant improvements in sparsity and reconstruction quality. By constructing sparse representations that are highly tuned to the signal characteristics, data-driven dictionaries provide more efficient signal reconstructions, especially in challenging scenarios like arrhythmia detection. This approach is particularly beneficial for real-time applications where flexibility and adaptability are crucial.

The selection of the sensing matrix also plays a pivotal role in CS. Several studies, such as [104, 105], have introduced adaptive sensing matrices to dynamically adjust the compression quality based on network conditions, optimizing real-time ECG data transmission and reconstruction. Moreover, various CS framework have been proposed, Polania *et al.* [138] enhance the accuracy by using iterative hard thresholding (IHT) and compressive sampling matching pursuit (CoSAMP) algorithms by leveraging prior knowledge of signal structure. Despite these advancements, challenges remain in balancing the trade-offs between real-time performance, power consumption, and signal quality. Another study [139] have explored real-time reconstruction using iterative shrinkage-thresholding algorithms (ISTA), but further investigation is needed into optimizing execution time and resource allocation in portable health monitoring systems. Additionally, several sensing matrices used in literature to enhance the compression performance, such as the RGM [140], Random Sparse Binary Matrix (RSBM) [138], and structured Fourier sensing matrices [141]. These studies demonstrated the critical impact of matrix selection on reconstruction accuracy.

CS has demonstrated considerable promise in reducing data acquisition and computational requirements for ECG monitoring systems, several areas still require further exploration. The integration of adaptive sensing matrices, data-driven dictionaries, and energy-efficient reconstruction algorithms remains a key challenge for achieving real-time, secure, and accurate ECG signal processing in portable health monitoring devices. This chapter aims to address these challenges by proposing data driven sensing matrix and overcomplete sparse dictionaries. Additionally, the proposed sensing matrices have been tested from hardware for their energy efficiency.

4.3 Sensing Matrices for ECG Signals

The CS recovery phase only guarantees high quality reconstruction when CS compression phase follow certain essential conditions, described in details in section 2.5. The sensing matrix is one of the most important component for these conditions such as RIP condition to guarantee the significant recovery of the sparse ECG signals. But only sensing matrices designed from random Gaussian or Bernoulli indices can satisfy the RIP and difficult to design the deterministic or structured matrices that follow the RIP. In this section, we will describe the designing random, and data-driven sensing matrix for ECG signal compression.

4.3.1 Random Sensing Matrices

Random sensing matrices are widely used in CS for ECG signals to project a high-dimensional signal onto a lower-dimensional space while preserving the essential information required for signal recovery [38, 100, 101]. Because such matrices are chosen to meet the CS essential conditions such RIP and also due to low coherence with sparse basis which untimely guarantee the signal reconstruction from compressed measurements. The most common types of random sensing matrices are Random Gaussian Matrix (RGM), Random Bernoulli Matrix with binary and signed entries.

Random Gaussian Matrix (RGM)

Each entry Φ_{ij} in the matrix is drawn independently of a Gaussian (normal) distribution with zero mean and variance $1/M$, where M is the number of measurements. The matrix is defined as follows:

$$\Phi_{ij} \sim \mathcal{N}(0, \frac{1}{M}), \tag{4.1}$$

where Φ is the $M \times N$ sensing matrix, M is the number of rows (measurements), and N is the length of the signal. Gaussian random matrices are known to satisfy the RIP with high probability, which attract the researchers for best choice for signal recovery in most of CS applications.

Random Bernoulli Matrix with Binary Entries

A *Sparse Binary Matrix* is another form of sensing matrix, which is generated using Bernoulli trials. In this case, each entry Φ_{ij} of the matrix is a binary value (0 or 1), determined by the probability p of a '1' occurring. The matrix is defined as follows:

$$\Phi_{ij} = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p, \end{cases} \tag{4.2}$$

where p is the probability of an entry being 1. In practice, a small value of p, such as $p = 0.1$, is often used to ensure sparsity in the matrix, which is beneficial in hardware implementations.

Random Bernoulli Matrix with Signed Entries

A *Random Bernoulli Matrix with signed entries* is an extension of the sparse binary matrix, where each entry Φ_{ij} is either $+1$ or -1, instead of just 0 or 1. The matrix is defined as:

$$\Phi_{ij} = \begin{cases} 1 & \text{with probability } p, \\ -1 & \text{with probability } 1-p, \end{cases} \tag{4.3}$$

where p is the probability of a $+1$ entry. This type of matrix maintains the randomness of Bernoulli trials but introduces signed entries, which can further improve the sensing matrix's incoherence with the sparse basis.

Each of these random matrix types plays a crucial role in the measurement process for compressed sensing. While the RGM offers theoretical guarantees through the RIP, the RBM is more efficient in terms of memory and computational cost, particularly in hardware implementations. The Random Bernoulli Matrix with signed entries provides a balance between randomness and low coherence, offering good performance with reduced complexity.

In practice, the choice of matrix depends on the specific application and constraints. For instance, hardware implementations often prefer sparse or binary matrices due to their simplicity, whereas applications that prioritize accuracy and recovery performance may opt for Gaussian random matrices.

4.3.2 Data-Driven Sensing Matrix

A data-driven sensing matrix is a sensing matrix that is learned or optimized directly from data, rather than being constructed based on random distributions (like Gaussian or Bernoulli) or deterministic rules (like Fourier or Hadamard matrices). The goal of a data-driven sensing matrix is to adapt the sensing process to the specific characteristics of the data, thereby improving the accuracy of signal recovery in CS.

The data driven sensing matrix include following key features such as data adaption, learning from the data and optimization. Unlike random or deterministic sensing matrices, these matrices designed to exploit the patterns, specific correlation or important features presents in the data. Additionally data driven matrices are learned from optimization techniques such as deep learning, machine learning or some specific optimization algorithms to minimize the reconstruction error and maximize the signals signal features acquisitions. In this research we employed simple machine learning optimization approach to design the data-driven sensing matrix to maximize the signal reconstruction quality and minimize the error. The fig 4.1 shows the overall proposed methodology for designing the sensing matrix and conversion into binary matrix for hardware implementation. The main idea of adopting this methodology is to design the sensing matrix

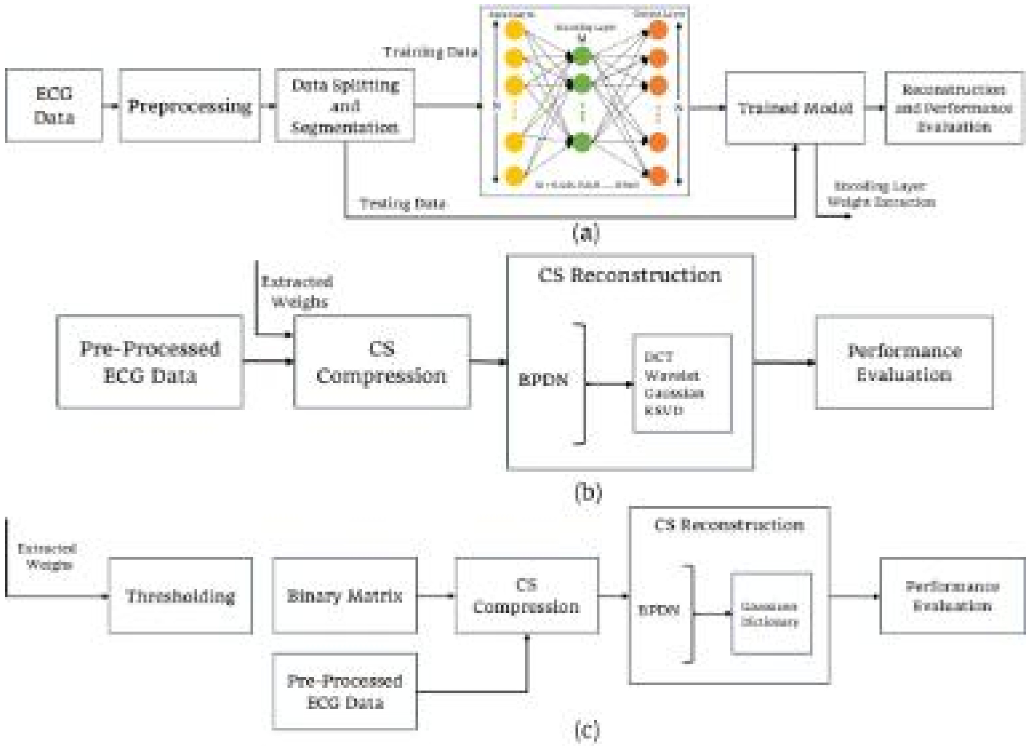

Figure 4.1: Proposed Method for Designing Data-Driven Sensing Matrix

which consume less resources when to be implemented in hardware, there is only one way to design the binary matrix which probably consume less resources in hardware. The figure 4.1(a) show the generating the data driven sensing matrix from ML optimization model. We used the MIT Arrhythmia data set described in 4.6.2 which consist of 48 records, data were pre processed such as removing the noise from the raw data and splitted into the training and testing set. We used 38 records for training and 10 for testing. Every training and testing record set is further divided into segments with fixed segment length of $N = 256$ samples. The Neural Network model was designed with 3 layers, input features layer with size of N, encoding layer with size of M (number of measurements) and decoding layer with size of N. Model was trained with different size of encoding layer $M = [25, 51, 76, 102, 128, 153, 179, 204, 230]$ selected with respect to compression ratio $CR = [90\%, 87\%, 83\%, 80\%, 70\%, 60\%, 50\%, 40\%, 30\%, 20\%, 10\%]$. The trained models at all compression ratios was evaluated on test data and figure 4.2 shows testing PRD error for all compression ratios. The plot shows that PRD increases with compression ratio, this trend indicates that as the compression ratio becomes higher, the reconstruction quality declines, which is expected since more information is lost when fewer measurements are retained. At lower compression ratios, the PRD remains relatively low, suggesting good reconstruction quality, but the PRD begins to rise sharply after around 60%, peaking at 90%. From the CR 10%-83%, the PRD remains under 9%, indicating a good quality of reconstruction. However, afterward, the PRD begins to increase more significantly.

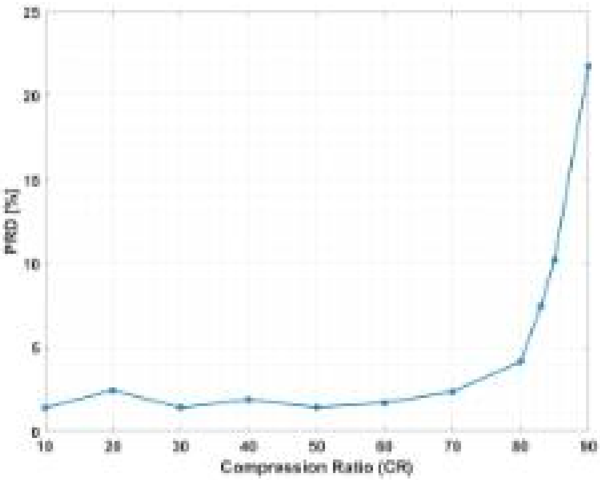

Figure 4.2: Models Testing Error at Different Compression Ratios

The Fig 4.1(b) shows the process of weight extraction from the trained models. The encoding layer weights from the trained models was extracted and treated as data driven sensing matrix (DSM). For instance, in the later phase the signal was compressed with this DSM. To comprehensively evaluate the performance of proposed data driven sensing matrix we used BPDN reconstruction algorithms with over-complete sparse dictionaries (Gaussian and K-SVD) and fixed standard Sparse Dictionaries (DCT and DWT). Considering the hardware implementation phase in mind, it's difficult to implement the proposed data-driven sensing matrix because the elements of the matrix are in floating point or decimal numbers and thus consume more memory and resources such as multiplication and addition operations. Therefore, we proposed novel thresholding technique to convert the floating point elements of the matrix to the binary matrix. Figure 4.1(c) shows the CS framework for binary thresholding matrix. Additionally the details of the thresholding are given in the Algorithm 3.

Algorithm 3 Thresholding approach for generating Binary Sensing Matrix

INPUT: Floating-point sensing matrix $\Phi \in \mathbb{R}^{M \times N}$, desired density levels D
OUTPUT: Binary sensing matrix $\Phi_{\text{BTM}}^{M \times N}$, best threshold value T_{best}, best density percentage D_{best}, and best PRD value

Initialize:
$\Phi_{\text{values}} \leftarrow$ Flattened list of all elements in Φ
Precompute thresholds T based on desired density levels S
for each density level $D_i \in \{D_1, D_2, \ldots, D_n\}$ **do**
    Compute threshold T_i such that $D_i\%$ of $\Phi_{\text{values}} \geq T_i$
    Apply threshold T_i to Φ to create binary matrix Φ_{BTM}:

$$\Phi_{\text{BTM}}(i,j) = \begin{cases} 1 & \text{if } \Phi(i,j) \geq T_i \\ 0 & \text{otherwise} \end{cases}$$

    Compute actual density $S_{\text{actual}} = \frac{\text{number of ones in } \Phi_{\text{BTM}}}{M \times N} \times 100$
    Perform compressed sensing (CS) using Φ_{BTM}
    Use reconstruction algorithm with desired sparse base to reconstruct the signal and compute PRD
    Store S_{actual}, PRD, and T_i for this threshold T_i
end for
Find the minimum PRD from the stored PRD values
Select the corresponding density level D_{best} and threshold T_{best}
Generate the optimized binary sensing matrix $\Phi_{\text{BTM}}^{M \times N}$ using T_{best}

4.4 Sparse Representation of ECG signals

This section, we will discuss how ECG signals can be sparsely represented to improve both compressibility and reconstruction quality. We begin by defining standard sparse dictionaries, followed by a discussion of the overcomplete sparse dictionaries. The benefits and limitations of each approach are also described at the end of this chapter.

4.4.1 Standard Sparse Dictionaries

As discussed before, selecting an appropriate sparse dictionary is critical for achieving high-quality signal reconstruction, especially at high compression ratios.

However, standard dictionaries such as Wavelet or DCT tend to exhibit limited sparsity when applied to ECG signals [142]. This has led many researchers to propose signal-specific or patient-specific dictionaries for better performance. For instance, Maiorescu et al. [143] demonstrated that temporal EEG signal and channel-specific dictionaries significantly outperform standard wavelet dictionaries in terms of reconstruction accuracy. ECG signals are not sparse in the time domain, but they do exhibit sparsity in certain transform domains. The wavelet transform typically provides a relatively sparse representation of ECG signals. Numerous wavelet families are available, and it is important to compare them to determine which is most suitable for ECG signal reconstruction. In our analysis in previous chapter, we evaluated multiple wavelet families and found that the ECG signals exhibit the highest sparsity with the *rbio5.5* wavelet. Several studies [53], [144], [145] have also explored the use of the standard DCT as a sparse basis, yielding promising results. While Wavelet, Fourier, and DCT are all effective sparse bases for ECG signals, a large number of active components may still be present, which undermines the sparsity required for optimal CS performance. To address this limitation, researchers have introduced signal specific and overcomplete dictionaries that offer improved sparsity for ECG signals. In this regard, M. Rakshit et al. [59] proposed a beat-type dictionary that provides high-quality signal recovery for individual ECG recordings without the need for extensive training. Similarly, D. Craven et al. [137] employed an adaptive dictionary scheme using multiple dictionaries generated via the dictionary learning (DL) technique, which further enhances signal reconstruction performance.

Standard dictionaries do not provide sufficient sparsity for ECG signals, prompting many researchers to propose overcomplete dictionaries to improve signal reconstruction and sparsity. However, most overcomplete dictionaries found in the literature are either patient-specific or signal-specific. In this work, we present two overcomplete dictionaries called Gaussian and Data-Driven K-SVD Dictionaries which are not patient or signal specific dictionaries and also shows the highest sparsity for ECG signals, making them more suitable for CS framework.

4.4.2 Mathematical Model of ECG Signal

We adopt a model for ECG signals that relies on a fundamental class of mathematical functions, facilitating the design of an effective dictionary representation based on this model. This model is derived from the Gaussian wave dynamical system initially proposed by McSharry *et al.* [146], which is employed to generate realistic synthetic ECG signals. The key idea is to represent the defining features of the ECG signal (i.e., the P, Q, R, S, and T waves) using Gaussian functions, with each wave characterized by three parameters: amplitude, width, and phase.

Mathematically, a one-dimensional Gaussian function is expressed as follows:

$$g_{p,b}(n) = a \exp\left(-\frac{(n-p)^2}{2b^2}\right) \tag{4.4}$$

where a represents the amplitude, p shows the position or phase, and b represents the shape or width of the Gaussian function. This allow to capture the morphology of a complete ECG cycle using an ordinary differential equation:

$$\dot{z}(a_i, b_i, p_i) = -\sum_i a_i \Delta p_i \exp\left(-\frac{\Delta p_i^2}{2b_i^2}\right) \tag{4.5}$$

where $\Delta p_i = (n - p_i)$ represents the relative position, a_i denotes the amplitude, and b_i corresponds to the shape or width of the i-th wave.

This model has been shown to be effective in tasks such as ECG signal compression, filtering, and classification. To fit an original ECG cycle $x(n)$ with the model described in Eq. 4.5, one can solve the following non-linear least-squares optimization problem:

$$\min_{a_i, p_i, b_i} \|x(n) - z(n)\|_2^2 \tag{4.6}$$

where

$$z(n) = \sum_i 2a_i \Delta p_i \exp\left(-\frac{\Delta p_i^2}{2b_i^2}\right) \tag{4.7}$$

The number of Gaussian functions required to represent a typical noiseless ECG cycle is a useful measure of the model's sparsity. According to prior research in [147], symmetric waves like the Q, R, and S waves can generally be approximated by a single Gaussian function each, while the asymmetric P and T waves may require two Gaussian functions. Consequently, an ECG cycle can be approximated using approximately seven Gaussian functions.

We have approximated the ECG signal using a linear combination of five Gaussian functions, each with different shape parameters: a for amplitude, p for position or phase, and b for shape or width. In practice, the real ECG signal is often mixed with various sources of noise. Therefore, the model discussed here is only an approximation of the ECG signal, as shown in figure 4.3.

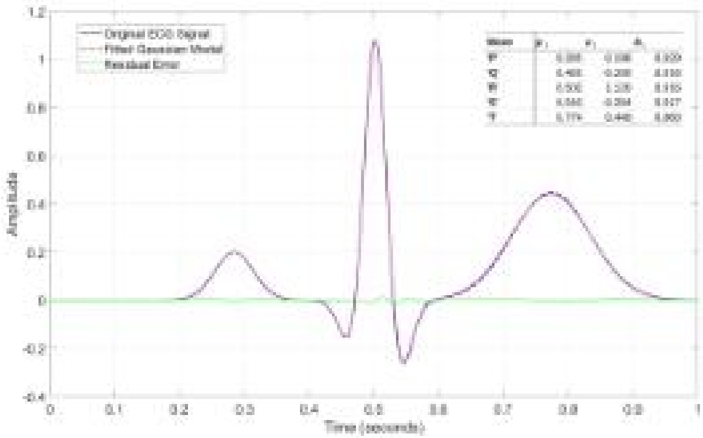

Figure 4.3: ECG approximated signal with 5 Gaussian function with 360 sampling frequency

4.4.3 Gaussian Sparse Dictionary Design for ECG Signal

As discussed in the previous section, most researchers have used fixed dictionaries as the sparse basis for signal compression. Some other authors have employed data-driven dictionaries [148], but such sparse bases are not universal. They either depend on specific heartbeats or require synchronization and normalization of the segments to be compressed. To address these issues, this work present a universal Gaussian dictionary based on the framework defined in [149] that does not require any training procedure, is independent of specific heartbeat types, does not rely on patient-specific signals, and is capable of representing abnormal patterns.

In the designed dictionary matrix Ψ, where each column represents an atom derived from samples of a Gaussian function:

$$g_{a_i,p_i}(t) = e^{\left(\frac{t-p_i}{a_i}\right)^2} \tag{4.8}$$

The parameters p_i and a_i are chosen appropriately as shown in figure 4.4. Various strategies were explored by testing different sets of scale parameters

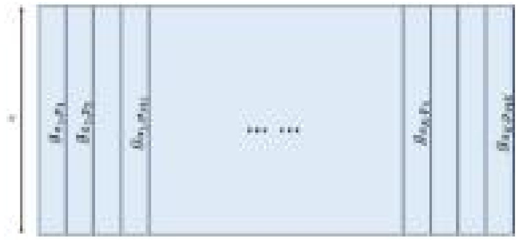

Figure 4.4: Gaussian Overcomplete Dictionary

and numbers of atoms. For example, assuming X is an ECG sample vector of length $N = 256$, the atoms are generated using scale parameters $a_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 49, 51\}$.

For each scale parameter a_i, all possible shift parameter values p_i within the vector are considered. Since the sample segments $N = 256$ are not synchronized with heartbeats, atoms near segment boundaries are created using windowed Gaussian functions. The reconstruction is performed at the receiver side based on the m measurements obtained by the acquisition device, where computational complexity and storage demands are less critical.

To assess the dictionary's ability to sparsify ECG signals, we examine the compressibility of ECG data using this dictionary. We compute the approximation coefficients with the Orthogonal Matching Pursuit (OMP) algorithm and sort them. Sorting the coefficients s_i such that $|s_1| \geq |s_2| \geq \cdots \geq |s_K|$, we observe a power-law decay. Figure 4.5(a) shows the absolute value of the sorted coefficients on a logarithmic scale, averaged across all ECG records from the MIT-BIH Arrhythmia Database [124, 125]. The faster the decay of coefficients, the more compressible the signal. As shown, the sparse representation of the ECG using the Gaussian dictionary exhibits this power decay law.

Next, we compare the approximation signal-to-noise ratio (SNR) between a fixed basis and proposed dictionaries by varying the number of retained coefficients. Figure 4.5(b) plots the approximation SNR against the sparsity factor S_f, defined as the ratio between the number of non-zero coefficients in the sparse representation and the original signal length $S_f = \frac{k}{N}$. The results indicate that the Gaussian and K-SVD outperforms the DWT (*rbio5.5*) and DCT at lower values of S_f, allowing for a sparser approximation. Consequently, fewer measurements are needed in the CS compression stage to achieve better reconstruction quality.

4.4.4 Data-Driven Overcomplete Dictionary Design for ECG Signals

A data-driven overcomplete dictionary can be designed using the K-SVD algorithm [150]. K-SVD is one of the most popular methods for dictionary learning (DL), as it adapts the dictionary atoms to fit the signal structure through a data-driven approach. This section details the design, training, and application of an overcomplete dictionary for ECG signals.

An overcomplete dictionary refers to a dictionary where the number of atoms (dictionary elements) exceeds the dimensionality of the signal. This redundancy allows for a more flexible and accurate representation of complex signals. For signals like ECG, which are inherently non-stationary and exhibit a wide variety of waveforms, including P-waves, QRS complexes, and T-waves, an overcomplete dictionary provides the adaptability needed to capture these variations. By using

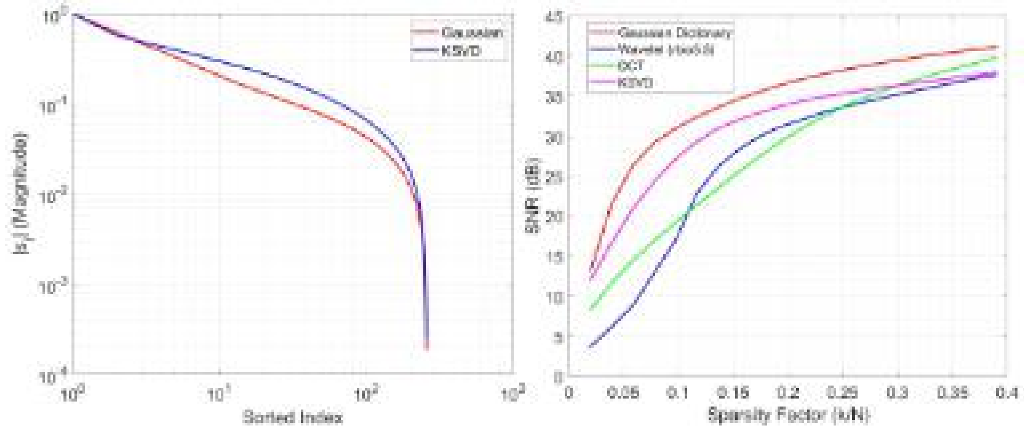

Figure 4.5: (a) Power decay curves for the sparse representation of ECG using the Gaussian dictionary; (b) Average SNR for various sparsity factors.

a larger set of atoms, the signal can be represented with more precision, leading to improved sparsity in the compressed sensing framework.

The K-SVD (K-means Singular Value Decomposition) algorithm is a dictionary learning method designed to find sparse representations of signals. It simultaneously updates both the sparse codes and the dictionary atoms. The learning process alternates between two stages, sparse coding and dictionary update.

Sparse Coding Phase: Given the input signal matrix X (where each column is a signal), the sparse coding phase involves representing each signal as a linear combination of dictionary atoms with as few non-zero coefficients as possible. This can be formulated as:

$$\mathbf{X}_i \approx \mathbf{\Psi}\mathbf{s}_i, \quad \text{subject to} \quad \|\mathbf{s}_i\|_0 \leq L$$

where:

- $\mathbf{X}_i$ is the i-th signal from the dataset $\mathbf{X}$,
- $\mathbf{\Psi}$ is the current dictionary (initialized randomly),
- $\mathbf{s}_i$ is the sparse coefficient vector,
- L is the sparsity constraint, ensuring that only a limited number of non-zero entries in $\mathbf{s}_i$ are allowed.

To solve this, we used Orthogonal Matching Pursuit (OMP). OMP iteratively selects the dictionary atoms that best match the signal, ensuring that the reconstruction error is minimized. The optimization at this stage can be expressed as:

$$\mathbf{s}_\Omega = \arg\min \|\mathbf{X}_i - \mathbf{\Psi}_\Omega \mathbf{s}_i\|_2$$

where Ω is the set of selected atoms in the dictionary that participate in the sparse representation of the signal. The residual is updated at each iteration:

$$\mathbf{r} = \mathbf{X}_i - \mathbf{\Psi}\mathbf{s}_i$$

This process continues until the sparsity level L is reached.

Dictionary Update Phase: Once the sparse representations (i.e., $\mathbf{s}$) of all the signals have been computed, the dictionary atoms need to be updated. For each atom ψ_j in the dictionary, the signals that use this atom are identified. The goal of the dictionary update phase is to adjust the atom ψ_j to better fit the signals, while minimizing the reconstruction error. The residual excluding the contribution of atom ψ_j is computed as:

$$\mathbf{E} = \mathbf{X} - \mathbf{\Psi}\mathbf{s} + \psi_j \mathbf{s}_j$$

Here, $\mathbf{E}$ is the residual matrix, and $\psi_j \mathbf{s}_j$ represents the contribution of atom ψ_j to the signals using it.

To update the atom, Singular Value Decomposition (SVD) is applied to the residual matrix:

$$\mathbf{E}_\Omega = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

The dictionary atom ψ_j is then updated to the first left singular vector:

$$\psi_j = \mathbf{U}_1$$

The sparse coefficients corresponding to this atom are also updated based on the first singular value:

$$\mathbf{s}_j = \mathbf{S}_1 \mathbf{V}_1^\top$$

This process is repeated for each atom in the dictionary until all atoms are updated. After each iteration of the sparse coding and dictionary update phases, the reconstruction error is evaluated to assess the accuracy of the signal approximation. The reconstruction error is defined as the Frobenius norm of the difference between the original signal matrix $\mathbf{X}$ and its approximation $\mathbf{\Psi}\mathbf{s}$:

$$\text{Error}(t) = \|\mathbf{X} - \mathbf{\Psi}\mathbf{s}\|_F^2$$

The algorithm continues to iterate until the stopping criterion is met, which is defined as:

$$\text{Error}(t) < \epsilon \quad \text{or} \quad t = T_{\max}$$

where ϵ is the predefined error tolerance and $T_{\max}$ is the maximum number of iterations.

The final learned dictionary, which contains more atoms than the length of the signal, is called an overcomplete dictionary. This overcompleteness provides greater flexibility in the sparse representation of signals, allowing the signal to be represented with fewer non-zero coefficients. This characteristic is crucial in improving the quality of signal reconstruction in the compressive sensing framework.

4.5 Reconstruction Algorithms

In this section we will describe the most commonly used reconstruction algorithms CS, focusing on their application to ECG signals.

4.5.1 Basis Pursuit Denoising (BPDN) Algorithm

In case of noiseless conditions, the sparse recovery problem shown in Eq.2.8 is non-convex optimization problems, and as mentioned before, it is considered as NP-hard. These problems can be approximated using convex relaxation, resulting in the Basis Pursuit (BP) problem described in Eq.2.9.

As a matter of fact, we can rewrite s as $s = u - v$, where $u, v \in \mathbb{R}^N$ are non-negative vectors. The vector u holds the positive components of s, while v captures the absolute values of the negative components of s.

The new vector $z = \begin{bmatrix} u^T & v^T \end{bmatrix}^T \in \mathbb{R}^{2N}$, which concatenates the vectors u and v. Now, the l_1-norm of s can be expressed as: $\|s\|_1 = 1^T(u + v) = 1^T z$ and the product Θs can be written as: $\Theta s = \Theta(u - v) = \begin{bmatrix} \Theta & -\Theta \end{bmatrix} z$. Therefore, we can recast the Basis Pursuit (BP) problem as a linear programming (LP) problem:

$$\min_z 1^T z \quad \text{subject to} \quad y = \begin{bmatrix} \Theta & -\Theta \end{bmatrix} z, \quad z \geq 0. \tag{4.9}$$

This LP can be solved by using well-established numerical algorithms, like interior-point methods [151].

In the presence of noise or when the signal is not exactly sparse, stability is ensured by solving the extended version of the previous problem, leading to Problem 2.12. This problem is known as quadratically constrained basis pursuit, where the constraint relates to the noise level in the solution. The basis pursuit denoising (BPDN) algorithm can be used to solve the second-order cone programming (SOCP) problem in Eq.2.12 by converting it into the unconstrained convex problem:

$$\min_s \|s\|_1 + \lambda\|\Theta s - y\|_2^2 (BPDN), \tag{4.10}$$

where $\lambda \geq 0$ is a parameter related to the signal power.

When it is difficult to estimate the noise power to set a value for λ, and it is easier to quantify a bound on the signal's sparsity, the LASSO (Least Absolute Shrinkage and Selection Operator) formulation can be used:

$$\min_s \|\Theta s - y\|_2^2 + \eta\|s\|_1 (LASSO) \tag{4.11}$$

where η is a Lagrange multiplier. It is easy to show that for an appropriate value of η, the solution to LASSO is equivalent to the solution of the unconstrained BPDN problem.

Results about the robustness and stability of l_1-based methods for signal reconstruction are mostly dependent on the properties of the matrix Θ. As discussed in chapter 2, if the matrix Θ satisfies the RIP, reconstruction from noiseless data is exact. Similar results hold for stable recovery in noisy settings or for compressible signals.

4.6 Results and Discussion

4.6.1 Experimental Set Up

According to the CS-based compression process discussed in previous chapters, we have collected compressed measurements $y \in \mathbb{R}^M$ using sensing matrix $\Phi \in \mathbb{R}^{M \times N}$ from segment of original ECG signal x with length N. We evaluate the performance of proposed DSM and DTM and compared with RGM, RBMB and RBMS. The length of ECG segment $N = 256$ is selected in order to work compression scheme in real time. This segment length is less than the $1second$ at sampling frequency $fs = 360Hz$. In our CS framework, it does not required any pre processing such as synchronization or normalization. While transmission of compressed measurements, it may corrupted with some noise $\hat{y} = y + noise$. Thus, our aim is to reconstructed or approximate the signal $\hat{x}$ from corrupted compressed measurements $\hat{y}$. To validate the performance proposed sensing matrices (DSM and BTM) and dictionaries (GD and K-SVD), we used BPDN reconstruction algorithm. Additonally, we also used two standard dictionaries (DWT and DCT) to compare the performance. All the experiments were conducted on the laptop with in Intel(R) core(TM) i7-12700HX 2100Hhz, 16 Cores using MATLAB 2023b. Additionally, we evaluated the same quality parameters as discussed in the Chapter 3, section 3.3.4.

4.6.2 Dataset Description

To assess the performance of the proposed overcomplete dictionaries and sensing matrix, we utilize the MIT-BIH Arrhythmia Database [124, 125]. This publicly available dataset contains 48 half-hour two-channel ECG recordings, collected from 47 patients. The recordings were digitized at a sampling rate of 360 samples per second per channel, with an 11-bit resolution over a 10 mV range. We evaluate the method using all MLII lead signals from the database. For the experiments, the first 5-minute segment from each record is extracted. Specifically, the reconstruction quality is assessed across a variety of signals, encompassing different rhythms, wave morphologies, and both normal and abnormal heartbeats.

4.7 Performance Evaluation of the Data Driven Sensing Matrix

We evaluated the performance of the proposed data-driven sensing matrix (DSM) by compressing ECG signals at various compression ratios (CR), ranging from 10% to 90% and dictionaries were evaluated by exploiting their sparsities with BPDN algorithm. We used the MIT-BIH Arrhythmia dataset which consist of 48 signals, where each five-minute ECG signal, consisting of 108,000 samples, was divided into equal segments of 256 samples for CS framework. After recovering each segment, PRD error was calculated. The final results represent the average performance across all segments within each trace and across all traces extracted from the signals. Table 4.1 and figure 4.7 shows the summary of results obtained from proposed CS framework. The table and plot collectively highlight the performance of the BPDN algorithm using different sparse dictionaries (DCT, DWT, Gaussian, and K-SVD) for ECG signal reconstruction at various compression ratios (CRs). It can be observed that the Gaussian (GD) and DCT dictionaries consistently yield the lowest PRD values across most CRs, indicating superior signal quality. At all CRs (10%-90%), GD, KSVD, DCT dictionaries generally maintain PRD values within an acceptable range (upto 9%), signifying good reconstruction quality at the same time DWT provides good quality reconstruction upto 70% compression. The DCT and Gaussian dictionaries demonstrate the best robustness across compression levels, while the DWT provides higher PRD,

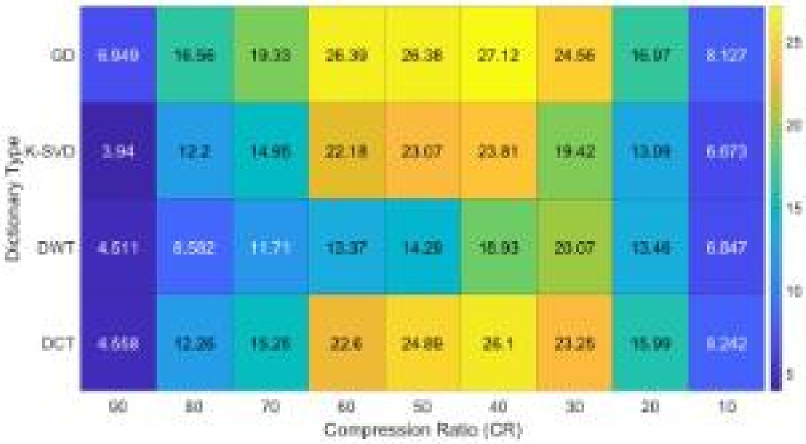

Figure 4.6: Quality Score Table for BPDN Algorithm

Table 4.1: Average PRD values for ECG reconstruction from CS measurements, exploiting sparsity in GD, K-SVD, DWT, and DCT.

Reconstruction Methods	**Sparse Dictionaries**	**Average PRD [%]**								
		CR [%]								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
BPDN	Gaussian	1.230	1.179	1.221	1.475	1.895	2.273	3.621	4.830	12.951
	K-SVD	1.499	1.528	1.544	1.680	2.168	2.705	4.683	6.556	22.844
	DWT	1.461	1.486	1.495	2.114	3.499	4.488	5.977	9.321	19.950
	DCT	1.213	1.251	1.290	1.532	2.009	2.655	4.590	6.527	19.744

particularly at higher CRs, indicating that DWT may be less suited for high compression scenarios. The overall trend confirms that BPDN with Gaussian dictionary is more effective for achieving low-error ECG reconstruction, especially at higher compression ratios. To provide the clear view, figure 4.6 reports the Quality Score heatmap which clearly indicate that GD achieve highest quality score across all compression ratios. It is clear that GD dictionary outperform the other sparse dictionaries and offer good quality reconstruction at higher compression ratio till 80%. Considering the hardware implementation point of view, it is very difficult to implement DSM on the hardware because it consist of decimal numbers thus it will consume more hardware resources. However, in the hardware low density binary matrices are easy to implement and also consumes less resources. Therefore, we proposed binary thresholding techniques to convert the DSM into binary thresholding matrix (BTM). The thresholding Algorithm is defined in algorithm 3. We will discuss the results obtained from Binary Thresholding Matrix in the next section.

4.8 Optimizing Binary Thresholding Matrix for Compressed ECG Measurements

The data-driven sensing matrix (DSM) was designed to improve the performance of compression and reconstruction. Using this DSM, a binary thresholding matrix (BTM) was generated through a thresholding approach (see the algorithm 3) aimed at optimizing the density level of the matrix. The optimization process involved varying the density levels from 0.5% to 12%, following the thresholding

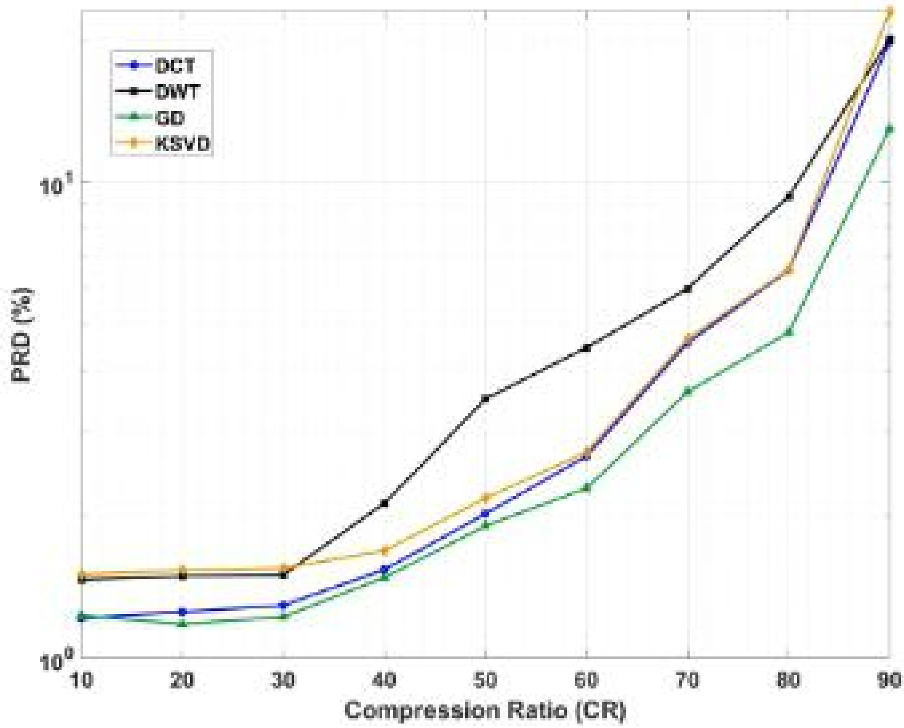

Figure 4.7: Performance Compassion of Different Dictionaries for CS Implementation

algorithm. The BTM was subsequently used to compress ECG signals, which were then reconstructed using the Basis Pursuit Denoising (BPDN) algorithm with Gasussian Dictionary.

4.8.1 Thresholding Process and Density Variation Impact

The thresholding process begins with the flattened DSM values and precomputes threshold values based on desired density levels D. For each density level D_i, a threshold T_i is calculated such that $D_i\%$ of the DSM values are above T_i, resulting in a binary matrix Φ_{BTM}. Each element in Φ_{BTM} is set to 1 if it meets the threshold condition $\Phi(i,j) \geq T_i$; otherwise, it is set to 0.

The impact of varying density levels on the BTM is significant, as it directly influences the sparsity of matrix. Lower density levels correspond to more zeros in Φ_{BTM}, leading to increased sparsity, while higher density levels maintain more of the DSM's structure, resulting in reduced sparsity as shown in figure 4.8 the BTM matrix at 50% compression ratio. This variation in density allows fine-tuning of the BTM for optimal compression and reconstruction, as increased density generally improves reconstruction accuracy but increases resource utilization.

4.8.2 Analysis of Reconstruction Error at Different Density Level of BTM

Figure 4.9 demonstrate the relationship between density of BTM and PRD error of reconstructed signal across various compression ratios. As observed, increasing the density generally leads to a decrease in PRD, indicating improved reconstruction quality. This trend is consistent across all compression ratios, with

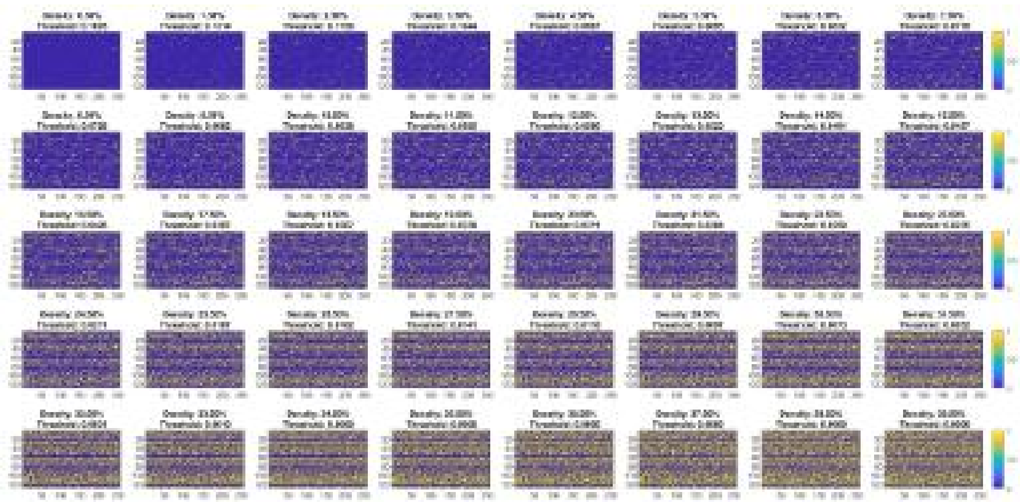

Figure 4.8: Binary Thresholding Matrix Optimization Example

higher densities yielding lower PRD values. The results show that for compression ratios up to 50%, increasing density stabilizes the PRD at lower levels, achieving a balance between sparsity and accuracy.

Specifically, the figure highlights that the minimum average PRD values are achieved at higher densities for higher compression ratios. For instance, at a compression ratio (CR) of 10%, the PRD stabilizes around a lower density level, while at a CR of 90%, a higher density is required to minimize PRD. This pattern reflects the trade-off between sparsity and accuracy, where higher compression ratios necessitate denser matrices to maintain reconstruction fidelity.

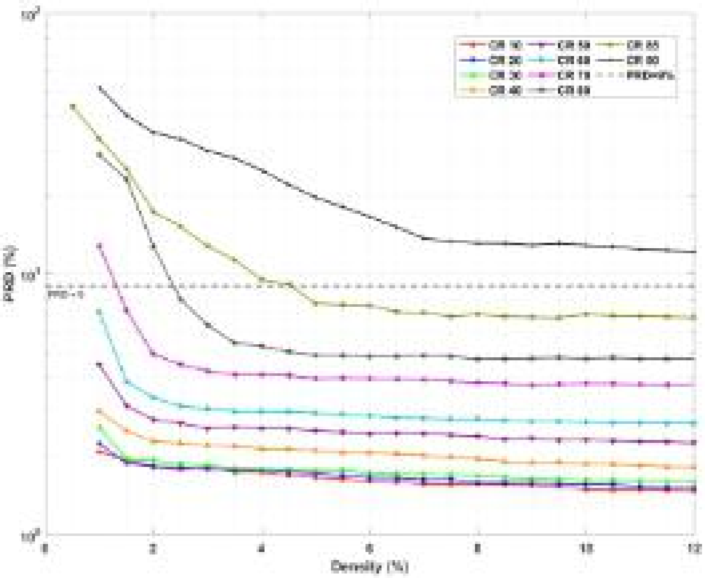

Figure 4.9: Density vs PRD trends at Different compression ratios of BTM

4.8.3 Finding the Optimal Density and Threshold Values

Table 4.2 provides a summary of the minimum PRD values, optimal density percentages, and corresponding threshold values across different compression ratios. For lower compression ratios (CR 10% to CR 50%), the best PRD values are achieved with density percentages close to 12%, indicating that denser matrices support better reconstruction. However, as the compression ratio increases to CR 70% and above, the optimal density levels gradually reduce, reaching around 8% for CR 80% and 9.5% for CR 85%. At CR 90%, the density returns to 12% to attain the lowest PRD.

Table 4.2: Minimum PRD values, optimal density percentages, and threshold values across all compression ratios

Compression Ratio [%]	**Minimum PRD [%]**	**Optimal Density**	**Threshold**
10	1.478	12.001	0.04397
20	1.514	12.000	0.04805
30	1.594	12.000	0.05251
40	1.825	11.499	0.05857
50	2.250	12.000	0.05763
60	2.675	11.500	0.06403
70	3.746	9.000	0.07191
80	4.727	7.996	0.07957
85	6.811	9.498	0.07045
90	12.210	12.000	0.05315

The results suggest that an adaptive approach to density selection, based on the compression ratio, can increase the ECG signal reconstruction quality. Lower compression ratios benefit from higher density matrices, which improve the fidelity of the reconstructed signal. Conversely, higher compression ratios require a delicate balance between sparsity and accuracy, with lower densities becoming favorable to reduce resource usage without significantly compromising PRD.

Finally, we provide a visual comparison of the reconstructed ECG signals in Figure 4.10. The plot shows the good reconstruction quality of various abnormal beats, with the original waveforms displayed in blue and the reconstructed waveforms in red. Each row corresponds to a different compression ratio (CR), ranging from 50% to 90%, and each column represents a distinct type of abnormal beat across two segments of four specific records. As observed in Figure 4.10, the reconstructed waveforms (red) closely overlap with the original waveforms (blue) across all types of abnormal beats. This alignment indicates that the essential characteristics of each waveform are preserved even after compression and reconstruction. The consistent overlay of the reconstructed signal over the original demonstrates the effectiveness of the proposed approach in retaining the distinguishing features of the ECG signals, essential for accurate clinical interpretation.

Benefits of Thresholding technique

The proposed thresholding method played a critical role in controlling the sparsity of the BTM, with varying densities significantly affecting the accuracy of ECG signal reconstruction. High-density matrices produced lower PRD values, es-

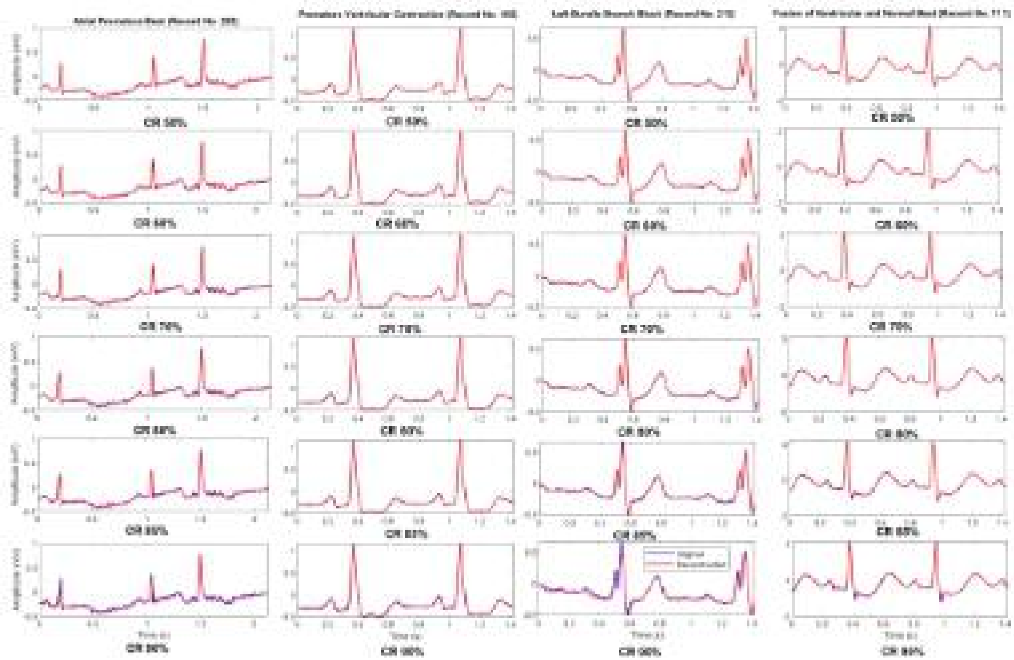

Figure 4.10: Original Signal (Blue) and reconstructed Signals (red) of few abnormal beats atrial premature beat (Record No. 222), premature ventricular contraction (Record No. 105),left bundle branch block (Record No. 213), and fusion of ventricular and normal beat (Record No. 111) extracted from the MIT-BIH arrhythmia database signals. Compressed with optimized BTM CR ranging from 50% to 90% and reconstructed with BPDN algorithm

pecially at lower compression ratios, due to increased matrix support for capturing most of the signal features. However, at higher compression ratios, maintaining high density led to diminishing returns in PRD reduction and increased resource utilization. This analysis indicates that optimizing threshold values to achieve an optimal density for each compression ratio is crucial for balancing reconstruction accuracy and computational efficiency. This approach allows for adaptable sparsity in the BTM, enhancing the effectiveness of compressed sensing for ECG signals, particularly in low-resource environments where efficient hardware utilization is necessary.

4.9 Comparative Analysis with State-of-the-art Sensing Matrices

This section presents a comparative analysis of proposed sensing matrix with state of the art sensing matrices (RGM, RBMB and RBMS), focusing on their performance in terms of compression ratio, reconstruction quality (PRD and SNR), sparsity, and hardware efficiency. Table 4.3 provides a summary of average PRD values of for each sensing matrix across different compression ratios (CR). The PRD values reflect the ECG signal reconstruction quality, with lower PRD values indicating better signal fidelity. Additionally the average PRD and SNR values acquired from our proposed sensing matrix BTM for all records are given in the table 4.4.

From the table 4.3, it is evident that the proposed DSM and BTM matrices generally outperform the traditional random matrices (RGM, RBMB, RBMS) across all compression ratios. Specifically, DSM achieves consistently low PRD values, maintaining high reconstruction accuracy even at higher compression ratios. BTM, although binary, also performs competitively with DSM, particularly at compression ratios below 60%, where it shows a marked improvement over RGM and RBMB. Due to the data driven and matrix structure to the statistical properties of ECG signals, the DSM and BTM, both matrices outperform the others. Unlike RGM, which is purely random, and RBMB/RBMS, which lack signal-specific optimization, DSM and BTM are capable of capturing ECG signal characteristics more effectively. This targeted design approach leads to better reconstruction of important signal features, resulting in lower PRD values and more accurate reconstruction.

Table 4.3: Comparative Analysis Table

Sensing Matrix	**Average PRD** [%]								
	CR [%]								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
RGM	1.208	1.839	2.191	2.379	2.561	2.871	3.644	5.619	17.136
RBMB	1.377	1.868	2.025	2.201	2.356	2.507	3.972	5.221	13.583
RBMS	1.323	1.803	2.149	2.333	2.486	2.581	3.974	4.938	13.131
DSM (Proposed)	1.230	1.179	1.221	1.475	1.895	2.273	3.621	4.830	12.951
BTM (Proposed)	1.444	1.479	1.539	1.714	2.052	2.406	3.467	4.464	11.881

4.9.1 Computational Complexity Analysis

Table 4.5 highlights the computational complexity and storage requirements of each matrix type. BTM's binary and sparse structure allows for simpler and faster operations, requiring fewer additions and no multiplications. In contrast, DSM, RGM, RBMB, and RBMS are less efficient for hardware. These traditional random matrices do not offer the same level of hardware efficiency or memory savings as BTM, as they are not optimized for binary or sparse processing. This distinction makes BTM particularly advantageous in settings where hardware constraints are significant, such as in wearable ECG monitors that prioritize power efficiency and speed.

4.9.2 Advantages and Limitation of Proposed BTM

The proposed binary thresholding matrix demonstrates significant results for ECG signals compression. Its binary structure, combined with the thresholding algorithm, allows BTM to achieve an effective balance between PRD, SNR, and computational efficiency. Unlike traditional random matrices, BTM is specifically optimized for ECG signals, ensuring high-quality reconstruction with minimal computational burden.

The use of BTM in hardware implementations offers more advanced benefits, including reduced memory and power consumption. This is particularly advantageous for portable and wearable ECG monitoring devices, where resources are limited. By providing a balance between reconstruction accuracy and hardware efficiency, BTM stands out as a viable solution for real-time, resource-constrained ECG monitoring applications.

Table 4.4: Average PRD and SNR values for all records of MIT BIH Arrhythmia Dataset

Record	**CR10**		**CR20**		**CR30**		**CR40**		**CR50**		**CR60**		**CR70**		**CR80**		**CR85**		**CR90**	
	PRD	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**	**PRD**	**SNR**
100	1.313	37.727	1.369	37.332	1.394	37.185	1.564	36.218	1.753	35.223	1.997	34.120	2.612	31.912	3.212	30.208	4.786	27.021	11.615	19.816
104	1.714	35.566	1.739	35.489	1.822	35.105	2.154	33.823	2.812	31.844	3.380	30.256	5.108	26.965	6.727	24.638	10.903	20.005	16.975	16.019
108	2.096	34.021	2.138	33.876	2.351	33.068	2.905	31.509	3.887	29.448	4.533	28.084	5.935	25.891	6.767	24.789	7.748	23.725	9.088	22.144
113	1.381	37.394	1.409	37.233	1.491	36.801	1.552	36.417	1.823	35.144	2.051	34.070	3.101	31.086	4.190	29.010	6.268	25.962	12.833	21.111
117	0.754	42.523	0.775	42.286	0.796	42.056	0.809	41.895	0.911	40.868	0.991	40.155	1.507	36.534	1.946	34.415	2.462	32.510	4.149	28.807
122	0.711	43.019	0.739	42.699	0.761	42.421	0.907	40.942	1.155	38.834	1.388	37.237	1.845	34.793	2.440	32.472	4.142	27.942	5.780	24.886
201	1.577	36.105	1.615	35.899	1.658	35.670	1.858	34.710	2.118	33.566	2.343	32.707	2.959	30.737	3.733	28.790	5.524	25.467	10.130	20.367
207	3.041	32.613	3.108	32.492	3.168	32.243	3.316	31.673	3.668	30.669	4.272	29.331	5.199	27.561	6.109	26.291	7.172	24.788	8.796	22.951
212	1.680	35.621	1.745	35.282	1.790	35.076	2.011	34.044	2.392	32.570	2.867	30.964	4.668	26.772	6.086	24.493	9.683	20.501	20.219	14.246
217	1.046	39.824	1.051	39.761	1.081	39.533	1.204	38.648	1.428	37.176	1.616	36.092	2.502	32.391	3.613	29.400	5.289	26.058	10.869	19.943
222	3.821	28.905	3.786	28.999	3.924	28.665	4.128	28.226	4.565	27.342	5.038	26.479	5.877	25.162	6.992	23.730	9.168	21.741	17.673	16.757
231	1.630	36.126	1.659	35.901	1.727	35.581	1.814	35.139	2.122	33.692	2.527	32.157	4.579	27.081	6.066	24.742	9.216	21.330	15.569	17.304
101	1.590	36.208	1.650	35.917	1.700	35.654	1.954	34.434	2.333	32.934	2.603	31.942	3.315	29.968	4.077	28.316	5.770	25.670	13.495	18.679
105	1.315	37.691	1.349	37.501	1.392	37.211	1.551	36.264	1.991	34.101	2.353	32.660	2.828	31.081	3.357	29.646	3.965	28.236	7.926	22.523
109	1.077	39.511	1.138	39.053	1.138	39.044	1.298	38.057	1.570	36.613	1.849	35.251	2.637	32.126	3.553	29.517	4.521	27.312	6.829	23.711
114	2.640	31.768	2.668	31.663	2.822	31.180	2.909	30.904	3.301	29.809	3.571	29.124	4.401	27.388	5.313	26.002	8.005	23.114	18.755	17.175
118	0.620	44.238	0.631	44.088	0.669	43.573	0.758	42.598	0.926	41.036	1.193	38.913	1.856	35.068	2.898	31.320	4.685	27.135	7.729	22.637
123	0.705	43.096	0.715	42.959	0.732	42.749	0.791	42.078	0.910	40.913	1.029	39.867	1.686	35.695	2.249	33.275	3.247	30.678	6.665	26.015
202	2.590	32.817	2.720	32.547	2.702	32.419	2.802	32.226	3.036	31.669	3.288	30.900	3.659	29.809	4.056	28.872	5.141	26.715	7.350	23.610
208	1.220	38.871	1.237	38.715	1.310	38.279	1.553	36.992	1.976	35.067	2.324	33.862	3.253	31.202	3.878	29.573	5.400	26.861	10.597	20.869
213	0.767	42.426	0.765	42.413	0.791	42.132	0.903	40.967	1.124	39.075	1.417	37.098	2.567	32.006	3.528	29.193	5.221	25.992	12.524	18.721
219	0.669	43.602	0.680	43.440	0.703	43.127	0.746	42.608	0.881	41.223	1.051	39.679	1.781	35.187	2.798	31.603	4.574	27.419	7.221	23.563
223	0.743	42.627	0.752	42.501	0.775	42.249	0.822	41.738	0.885	41.090	0.977	40.271	1.545	36.330	2.313	32.904	3.681	29.150	7.655	22.651
232	2.270	32.989	2.277	32.958	2.395	32.537	2.761	31.291	3.520	29.219	4.703	26.802	7.908	22.299	10.421	19.958	13.350	17.837	20.156	14.485
102	1.815	34.907	1.866	34.689	1.949	34.308	2.107	33.648	2.462	32.319	2.855	31.064	4.507	27.449	5.449	25.933	9.341	21.267	17.988	15.631
106	1.662	35.763	1.729	35.454	1.830	34.982	2.181	33.415	2.720	31.585	3.564	29.221	5.926	24.864	8.058	22.379	11.374	19.525	17.480	16.020
111	2.163	33.453	2.209	33.256	2.217	33.235	2.286	32.956	2.535	32.065	2.755	31.311	3.334	29.694	4.196	27.794	6.198	24.472	14.764	17.497
115	0.984	40.366	0.989	40.322	1.034	39.954	1.119	39.275	1.256	38.301	1.475	36.983	2.223	33.481	2.843	31.447	4.817	27.393	14.994	19.024
119	0.707	43.205	0.703	43.230	0.737	42.828	0.799	42.187	0.917	41.038	1.097	39.568	1.834	35.187	2.737	32.057	4.131	28.843	6.520	25.080
124	0.559	45.122	0.567	45.004	0.583	44.765	0.622	44.206	0.704	43.136	0.802	42.084	1.186	38.684	1.553	36.400	2.605	32.668	5.388	27.669
203	2.196	34.381	2.399	33.707	2.662	33.194	3.780	30.649	5.210	28.615	6.272	27.204	8.994	24.393	10.604	22.524	12.415	20.622	13.883	19.278
209	2.058	33.815	2.103	33.649	2.205	33.245	2.469	32.234	3.070	30.432	3.430	29.423	5.296	25.782	6.352	24.155	9.817	20.530	26.294	12.304
214	1.181	38.694	1.210	38.488	1.257	38.141	1.380	37.363	1.644	35.831	1.971	34.259	2.670	31.655	3.312	29.817	5.084	26.247	9.292	21.128
220	0.900	40.977	0.957	40.503	0.979	40.284	1.067	39.572	1.204	38.487	1.348	37.530	2.002	34.169	2.999	30.951	5.043	26.726	14.415	18.186
228	2.027	34.739	2.071	34.552	2.181	34.157	2.511	32.891	3.132	31.053	3.780	29.448	5.364	26.360	6.714	24.507	7.840	23.210	11.068	20.061
233	0.880	41.330	0.900	41.124	0.937	40.768	0.992	40.244	1.144	39.033	1.394	37.329	2.331	32.887	3.695	29.037	5.678	25.302	8.401	21.959
103	1.325	37.712	1.345	37.631	1.429	37.099	1.542	36.389	1.943	34.375	2.292	32.865	3.192	30.158	4.527	27.452	6.695	24.018	14.007	18.039
107	0.646	43.915	0.676	43.553	0.698	43.267	0.840	41.797	1.074	39.715	1.349	37.697	2.393	32.832	3.429	29.944	5.397	25.981	11.338	19.585
112	0.552	45.211	0.558	45.118	0.576	44.842	0.599	44.488	0.677	43.455	0.768	42.358	1.193	38.610	1.538	36.377	2.054	33.938	3.456	29.539
116	0.692	43.297	0.701	43.191	0.752	42.606	0.822	41.866	1.032	39.911	1.260	38.261	2.204	33.409	3.261	30.075	5.789	25.139	10.140	20.043
121	0.680	44.335	0.698	44.130	0.710	43.931	0.736	43.599	0.826	42.625	0.914	41.644	1.332	38.415	1.670	36.322	2.048	34.626	2.540	32.838
200	1.569	36.640	1.609	36.445	1.704	35.930	2.072	34.327	2.784	31.851	3.439	30.029	4.674	27.428	5.894	25.413	8.002	22.523	15.008	17.277
205	1.090	39.308	1.120	39.076	1.130	39.001	1.181	38.602	1.332	37.593	1.567	36.196	3.146	30.232	4.123	27.867	5.840	24.935	11.424	19.216
210	1.961	34.333	1.981	34.214	2.034	33.984	2.163	33.453	2.474	32.305	2.887	30.950	3.387	29.550	4.150	27.822	5.176	26.027	7.288	23.179
215	2.238	33.240	2.312	32.937	2.456	32.445	3.008	30.708	3.721	28.894	4.601	27.137	6.738	23.791	8.567	21.703	12.214	18.716	24.180	12.824
221	1.764	35.246	1.825	34.931	1.887	34.652	1.979	34.231	2.284	32.993	2.547	32.032	3.074	30.386	3.814	28.614	5.297	25.983	10.889	20.325
230	1.204	38.528	1.182	38.681	1.241	38.254	1.291	37.907	1.420	37.079	1.646	35.839	2.776	31.356	3.671	28.930	6.259	24.786	18.293	15.745
234	1.483	36.721	1.543	36.425	1.574	36.227	1.668	35.715	1.846	34.838	2.118	33.696	3.298	29.849	4.807	26.646	6.681	23.884	10.642	19.869
Average	1.444	38.136	1.479	37.944	1.539	37.618	1.714	36.773	2.052	35.347	2.406	34.003	3.467	30.743	4.464	28.486	6.369	25.428	11.881	20.444

Table 4.5: Sensing matrix computational operations and storage space allocation for ECG signal compression

Matrix	Add	Mul	Storage Space
RGM (Gaussian)	$(n-1)\times m$	$n\times m$	$n\times m$
RBMB (Bernoulli Binary)	$d\times(n-m)$	0	$n\times d$
RBMS (Bernoulli Signed)	$d\times(n-m)$	0	$n\times d$
DSM (Proposed)	$(n-1)\times m$	$n\times m$	$n\times m$
BTM (Proposed, Binary)	$d\times(n-m)$	0	$n\times d$

Additionally, we optimized the BTM for densities up to 12% across all compression ratios, which may provide more accurate results at higher densities. Another limitation of this matrix is that it is generated from the DSM, which was trained on the MIT-BIH Arrhythmia dataset. In the future, this approach could be further generalized by incorporating additional datasets.

4.10 Hardware Implementation

To evaluate the real-time feasibility of our optimized BTM for ECG signal compression, we implemented it on an STM32F401 Nucleo microcontroller, which operates at 84 MHz clock frequency with 512 KB of Flash memory and 96 KB of SRAM. We measured the processing time required to compress one complete segment of ECG data. Table 4.6 provides an overview of the performance metrics across various compression ratios, including the processing time, power consumption, and energy usage.

The STM32F401 processes ECG signals segment with an original sampling frequency of 360 Hz, meaning that a new sample arrives approximately every 2.78 milliseconds or 2780 microseconds. Given the processing times observed, which range from 34 to 218 microseconds depending on the compression ratio, the BTM implementation easily fits within this time window. In other words, the time required to compress each segment is significantly shorter than the interval between incoming samples, confirming that this implementation is well-suited for real-time applications.

Figure 4.11 illustrates the execution of the proposed BTM implementations on the microcontroller through simple timing diagrams. Given the 360 Hz sampling rate, a new sample is received every 2.77 ms, resulting in an initial latency of 2.84 ms before processing can be performed on the first segment. The microcontroller-based implementation requires an elaboration time, as shown in Table 4.6, which varies with the compression ratio. For all compression ratios, the proposed BTM completes all operations within 218 μs for $CR = 10\%$ and 34 μs for $CR = 90\%$ at an 84 MHz clock frequency. Once processing is complete, a new segment can be transferred to each implementation. This confirms that the microcontroller-based design is capable of sustaining real-time operation without the need for an additional buffer.

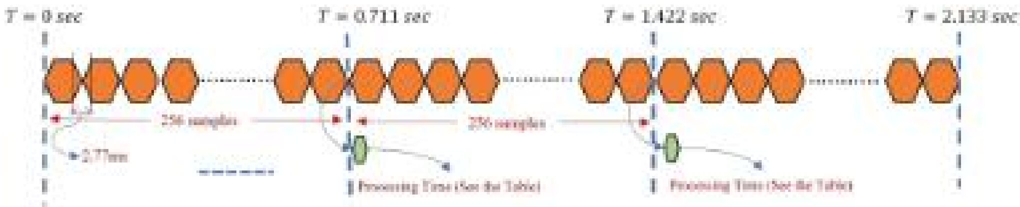

Figure 4.11: Timing diagram of Microcontroller Implementation

Table 4.6: Microcontroller Implementation Results (Current: 21.6 mA, Power: 71.28 mW, constant for all compression ratios)

Compression Ratio [%]	**Density**	**Average PRD**	**Processing Time (microseconds)**	**Clockcycles**	**Energy (µJ)**
10	12.0	1.444	218	18232	15.53904
20	12.0	1.479	193	16196	13.75704
30	12.0	1.539	171	14301	12.18888
40	11.5	1.714	141	11854	10.05048
50	12.0	2.052	124	10378	8.83872
60	11.5	2.406	95	7995	6.7716
70	9.0	3.467	58	4861	4.13424
80	8.0	4.464	38	3118	2.70864
85	9.5	6.369	34	2831	2.42352

4.11 Discussion

In this study, we developed and evaluated a data-driven approach to compressive sensing for ECG signals. Our proposed methods include a data-driven sensing matrix (DSM) and a binary thresholding matrix (BTM), both optimized for efficient ECG signal compression and accurate reconstruction. Using these matrices, we evaluated reconstruction quality across a range of compression ratios. Results demonstrate that our proposed BTM matrices yield high-quality reconstructions across varying compression ratios, achieving low PRD values and preserving the morphological features of the ECG signals. Specifically, the BTM consistently performed well even at high compression ratios (up to 85%), showing minimal signal degradation. This is particularly advantageous in real-time applications where data compression is essential to reduce power consumption, and channel datarate without compromising diagnostic quality. The preservation of critical waveform features in abnormal beats further suggests that our approach is suitable for clinical applications, as it can retain diagnostic information crucial for identifying cardiac abnormalities.

Compared to traditional random Gaussian and Bernoulli sensing matrices, our proposed BTM outperform these standard approaches, especially at higher compression ratios. Previous studies on compressive sensing for ECG signals have primarily focused on random matrices due to their simplicity and easy to follow the RIP. However, our findings indicate that data-driven matrices provide good reconstruction quality uplot 85% compression ratio. The BTM, in particular, addresses hardware implementation constraints by providing a sparse binary alternative that reduces computational complexity and hardware resource consumption while maintaining almost same reconstruction accuracy as compare to DSM. This advancement completely aligns with the goals of real-time implementation of CS for ECG signal monitoring in resource-constrained environment.

Considering the limitations of the this study, our DSM was specifically trained on the MIT-BIH Arrhythmia dataset, which, while widely used, may not encompass the full variability found in other ECG datasets or patient populations. This limitation may affect the generalizability of our findings to broader clinical settings. Additionally, although the BTM shows potential for hardware implementation, the thresholding process used to convert DSM into a binary matrix could potentially lead to information loss at extremely high compression ratios. The current study did not extensively explore the impact of high density which may consume more hardware resources but provide good quality reconstruction, which could further optimize the balance between sparsity and reconstruction accuracy.

4.12 Conclusion

In this chapter, we designed, implemented and evaluated a data-driven approach to compressive sensing for ECG signals, focusing on the development of a data-driven sensing matrix (DSM) and converting into a binary thresholding matrix (BTM). Our results demonstrate that the BTM, in particular, provides good quality reconstructions even at compression ratios as high as 85%, with minimal signal degradation, making it a suitable for real-time ECG monitoring in resource-constrained environments. Due to the binary and sparse structure of BTM offers significant advantages for hardware implementation by reducing computational complexity and power consumption without compromising diagnostic quality. When compared to traditional random matrices, our approach not only achieves superior reconstruction quality at higher compression ratio but also addresses real time constraints relevant to wearable and IoT health monitoring devices such as power consumption and datarate. However, the proposed sensing matrix reliance on the MIT-BIH Arrhythmia dataset for training may limit its generalizability, and future studies should investigate broader datasets to validate the robustness of our approach. Additionally, while BTM offers practical benefits, exploring alternative thresholding techniques and exolore higher density levels and optimal sparsity levels could further enhance the trade-off between resource efficiency and signal reconstruction quality.

Chapter 5

A Deep Learning Approach for ECG Compressed Sensing

5.1 Introduction

Electrocardiogram (ECG) monitoring has become increasingly essential in healthcare applications, providing a non-invasive method for early detection of heart diseases. ECG signals are commonly used in clinical diagnostics, including arrhythmia detection and evaluation of coronary artery blood flow. However, ECG data are typically acquired at high sampling rates, resulting in large datasets with extensive multi-lead recordings, which demand significant storage space. This creates a need for efficient solutions to reduce data transmission requirements, storage burden, and energy consumption.

Compressed Sensing (CS) offers a powerful approach to address these challenges by enabling joint signal acquisition and reconstruction. [152, 153]. By efficiently compressing ECG signals, CS reduces both the data volume and associated energy costs during transmission [77]. Once the compressed data is received, CS reconstructs the signals to a dimensionality similar to the original, making them viable for clinical diagnosis [154]. Traditional CS methods obtain measurements by multiplying the original signal by a known measurement matrix. Reconstruction from these measurements involves solving an underdetermined system, where success often relies on signal sparsity in a certain domain [155].

Key aspects of conventional CS recovery methods include:

- **Signal Sparsity**: Signals should either be inherently sparse or transformed into a sparse domain for CS to work effectively. Various transformations are commonly employed to achieve this, such as wavelet [77], Fourier [154], short-time Fourier [155], Beamlet [81], Curvelet [82], Contourlet [156], and K-SVD for dictionary learning [157].

- **Measurement Matrix Design**: An efficient measurement matrix should satisfy the Restricted Isometry Property (RIP) for accurate reconstruction. RIP is defined as:

$$(1-\delta)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta)\|x\|_2^2, \tag{5.1}$$

  where δ is a small constant. Measurement matrices with low correlation are more likely to satisfy RIP, allowing accurate signal recovery. Typical matrices include random (e.g., Gaussian, Bernoulli) and deterministic options, such as LDPC codes [158] and chaotic sequences [159].

- **Optimization for Reconstruction**: Non-convex optimization problems in CS are often reformulated as convex problems by adjusting the objective function or relaxing constraints.

5.2 Related Work and Motivation

Numerous studies have investigated CS for ECG applications, aiming to enhance compression and reconstruction quality. For example, Zhang et al. [160] introduced the Block Sparse Bayesian Learning (BSBL) algorithm, utilizing spatial and temporal structures to compress and reconstruct fetal ECG signals effectively. Izadi et al. [161] applied the Kronecker technique and adaptive dictionary learning to improve reconstruction, while Abhishek et al. [109] developed a biorthogonal wavelet based on double-sided exponential splines, achieving superior reconstruction quality. Other notable approaches include two-dimensional mapping for improved sparsity [162], incorporation of prior support information [163], and low-rank constraints that exploit spatial and temporal ECG signal structures [164]. Additionally, Saliga et al. [165] utilized a dynamic ECG model to reduce noise interference, while Rezaii et al. [166] proposed a method to optimize sparsity for noise robustness.

Despite the advantages of traditional CS, some limitations persist in practical applications. For instance, the measurement matrix often lacks structural information from the original signal, which can lead to reduced reconstruction quality, particularly at low sensing rates. Moreover, issues related to matrix storage, computational demands, and reconstruction speed hinder CS adoption.

With advancements in deep learning, CS has expanded into data-driven methods that enhance performance by learning signal characteristics directly from data. Deep neural networks, such as Convolutional Neural Networks (CNNs), replace traditional measurement matrices with learnable layers, enabling end-to-end optimization without requiring signals to be sparse. This adaptability has led to a range of deep learning-based CS applications [167, 168].

For example, Muduli et al. [169] introduced a Stacked Denoising Autoencoder (SDAE) to compress and reconstruct fetal ECG signals, while Polania et al. [168] employed Restricted Boltzmann Machines (RBMs) to capture ECG sparsity patterns, thereby enhancing accuracy. Mangia et al. [170] proposed a CNN with short windows to reduce computation complexity. Other methods include CNN-based CS frameworks for EEG signals [171], oracle-based decoders [172], and multi-layer perceptron networks trained with stochastic gradient descent [173].

Compared to traditional CS, deep learning-based methods exploit inter-data relationships to achieve higher reconstruction accuracy. Their end-to-end training removes the dependency on specific sparse representations, simplifying the process. However, maintaining high reconstruction performance at low sensing rates remains a challenge, motivating further research toward improving model design and quality. In this chapter, we used a data-driven sensing matrix for ECG signal compression and a lightweight neural network model for ECG signal reconstruction. Our contributions are listed below:

- We used a simple two-layer neural network model to train our data-driven sensing matrix. The neural network consists of two layers: an encoding layer and a decoding layer. After training the network, we extracted the weights of the encoding layer and designated them as the Data-Driven Sensing Matrix (DSM).
- We proposed a Binary Thresholding Matrix (BTM) and implemented it in hardware for real-time applications. The BTM was generated from the DSM using a thresholding approach.
- We introduced a lightweight, non-iterative, and fast reconstruction method for ECG reconstruction. The designed model was also implemented in hardware for real-time applications.
- We designed end-to-end sensing and reconstruction models for compressive sensing-based ECG signal acquisition and reconstruction. The model was implemented on hardware.
- We conducted experiments on the widely used MIT-BIH Arrhythmia Database to validate the proposed models. We evaluated metrics such as PRD, SNR, CR, and hardware resource utilization, including power, energy, and inference time.

5.3 Deep Compressive Sensing for ECG Signals

Recent advancements in deep learning have significantly impacted the field of CS for physiological signals, particularly in ECG signals. Deep learning based CS approaches offer an adaptive and data-driven method for efficiently compressing and reconstructing ECG data, which is essential for effective real-time health monitoring and resource-limited systems. The deep learning for compressive sesing focuses on two key components: End-to-End Training and Learnable Measurement Matrices.

5.3.1 End-to-End Training

Traditional CS methods approach signal compression and reconstruction as distinct stages, often requiring separate optimization steps and domain-specific transformations. In contrast, deep learning-based CS employs end-to-end training, where the entire process from compression to signal reconstruction is optimized simultaneously. This integrated approach allows a single model to learn the compression and reconstruction steps jointly, adjusting parameters to ensure the best possible result at each stage.

In end-to-end training, the machine learning or deep learning model is exposed to vast amounts of ECG data, allowing it to identify complex patterns and features that are critical to signal fidelity. Rather than relying on mathematical transformations like traditional CS, the model effectively learns how to compress and decompress the signals based on the data itself. This adaptive learning is similar to a holistic learning process, where each stage informs and refines the others. For instance, by training the model to recognize the recurring patterns of ECG signals, such as P, QRS, and T waveforms, it can focus on retaining the most diagnostically relevant information, even at reduced data rates.

5.3.2 Learnable Measurement Matrices

In traditional CS frameworks, measurement matrices are fixed, often selected based on theoretical properties rather than the unique characteristics of the target data. These predefined matrices may not fully capture the features of ECG signals, potentially leading to less accurate reconstructions. In deep learning-based CS, however, measurement matrices become adaptable, as they are treated as learnable parameters within the model. This flexibility allows the model to effectively design the custom or data driven measurement matrix tailored to the specific patterns and structures of ECG data during training. By using end-to-end training and learnable measurement matrices, deep learning-based CS models bring a flexible and highly effective solution to ECG signal compression and reconstruction.

5.4 Methods and Materials

In this chapter, we propose a light weight Neural Network based approach for the compressive sensing framework for ECG signals. This approach involves generating a data-driven sensing matrix, optimizing it into a binary sensing matrix for efficient signal compression, designing a lightweight neural network model for signal reconstruction, and implementing the model for both compression and reconstruction on hardware. The combined methodology aims to enhance the efficiency of signal compression while ensuring high-quality reconstruction, making it suitable for hardware implementation.

5.4.1 Generating a Data-Driven Sensing Matrix

Figure 5.1 illustrates the process of generating the data-driven sensing matrix. We used the MIT-BIH Arrhythmia dataset, as discussed in the previous chapter. The methodology includes several key steps: preprocessing, data splitting and segmentation, training the neural network (NN), evaluating the trained model and extracting the weights for the final data-driven sensing matrix.

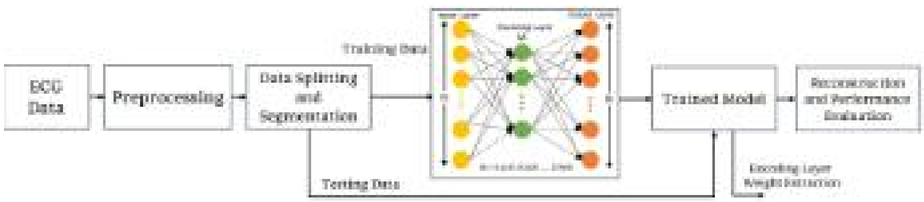

Figure 5.1: Process for Generating Data Driven Sensing Matrices

First, the ECG dataset were preprocessed to remove noise and artifacts. This step is essential for accurate training and reliable reconstruction performance. Next, the data is organized through data splitting and segmentation. Here, the ECG signals are divided into training and testing set as shown in the table 5.1, furthermore each records in both sets were segmented into fix segmemt length 256 samples, this approach will enable the model to learn more effectively from patterns within shorter windows.

Once the data has been segmented, it is fed into a neural network model for training. The architecture of this network consists of feature input layer, a encoding layer, and an output layer. The encoding layer, labeled as M in

Table 5.1: Training and Testing ECG Records in the Proposed Method

Training Set	**Testing Set**
100, 104, 108, 113, 117, 122, 201, 207, 212, 217, 222, 231, 101, 105, 109, 114, 118, 123, 202, 208, 213, 219, 223, 232, 102, 106, 111, 115, 119, 124, 203, 209, 214, 220, 228, 233, 103, 107	112, 116, 121, 200, 205, 210, 215, 221, 230, 234

Figure 5.1, represents the learnable matrix that will be optimized during training. The number of neurons in the encoding layer can vary as a fraction of the input dimension N, with values such as $0.1 \times N$, $0.2 \times N$, and up to $0.9 \times N$. This parameterization allows the model to experiment with different compression ratios.

After training the neural network, the model's performance is evaluated by measuring the PRD values of reconstructed signal from test dataset to ensure it meets the desired accuracy in reconstructing ECG signals. Finally, the weights of the first encoding layer are extracted to form the data-driven sensing matrix. These weights capture the essential features of the ECG signals, learned directly from the data, and are subsequently used as the sensing matrix for compressive sensing.

5.4.2 Proposed NN model for Compressive Sensing ECG Reconstruction

Figure 5.2 illustrates the proposed CS framework for reconstructing ECG signals. This framework integrates several key stages: data collection, preprocessing, segmentation, a neural network model for signal reconstruction, model testing, and performance evaluation. The process begins with ECG Dataset, where signals are obtained from MIT-BIH Arrhythmia database. Next, the noise and artifacts are removed to enhance signal quality. Preprocessing involve filtering techniques to remove first three harmonics of the power-line signal, at 60, 120, 180Hz frequencies respectively. resulting in a clearer signal representation.

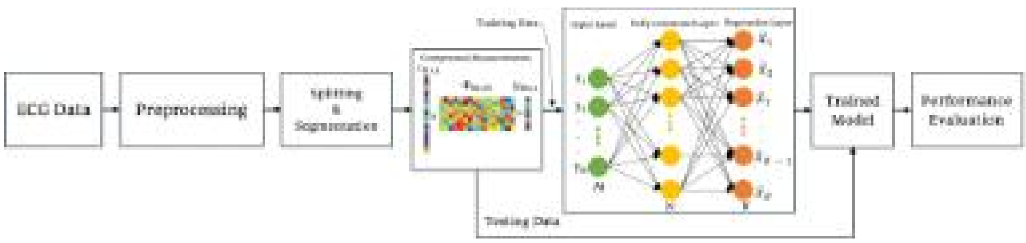

Figure 5.2: Proposed NN based CS framework for ECG reconstruction

Following preprocessing, the data is prepared for model training through Splitting and Segmentation. Here, the ECG signals are split into training and testing sets according to the table 5.1 and each record were segmented with fix segment size 256 samples and compressed with data driven sensing matrix generated in previous step.

The core of the framework is the Neural Network (NN) Model for CS Reconstruction. This neural network is structured with three main layers: an input layer, a fully connected layer, and a regression output layer. The input layer receives the compressed ECG signal. The fully connected layer represents the learnable structure that aids in reconstructing the compressed signal by capturing complex patterns inherent to ECG data. Finally, the regression output layer reconstructs the signal segments to their original form. The network is trained with Max Epochs = 200, Initial Learn Rate = 0.001 L_2 Regularization = 0.0001, batch size = 300, Learn Rate Drop Factor 0.5, Learn Rate Drop Period = 10 with the aim of learning an optimal mapping from compressed measurements back to high quality ECG signal representations. After training, the model is validated on test Dataset. Finally, the last stage is Performance Evaluation which inclues measuring PRD, SNR, reconstruction time.

5.4.3 Proposed Autoencoder Model for End-to-End Sensing and Reconstruction

Figure 5.3 shows the proposed autoencoder model, designed for end-to-end sensing and reconstruction within CS framework. The model shows an encoder-decoder architecture, where the encoder functions as a CS sensing matrix, compressing the input signals, and the decoder serves as a reconstruction mechanism to recover the original signal from the compressed measurements. This approach capitalizes on the inherent data-driven learning capabilities of neural networks to optimize both sensing and reconstruction processes jointly.

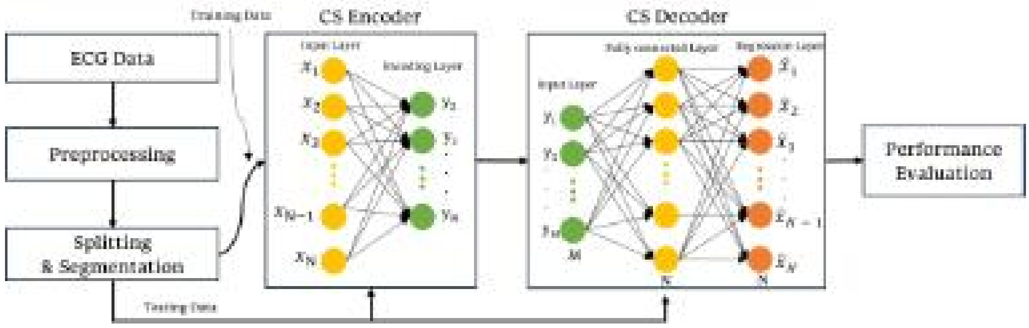

Figure 5.3: Proposed CS framework for End to End NN based Sensing and Reconstruction

The process begins with the input ECG data selection, preprocessing, splitting and segmentation stages which are the same as discussed in the previous sections. In the proposed model, the Encoder stage is responsible for compressing the ECG signals. The encoder consists of an *input layer* that receives the segmented ECG signal, followed by an encoding layer, which learns an optimal compressed representation of the signal. This encoding layer effectively functions as a data-driven sensing matrix, mapping the original input dimensions $X_1, X_2, ..., X_N$ to a lower-dimensional space $Y_1, Y_2, ..., Y_M$ (where $M < N$). The neural network learns this mapping during training, identifying key features in the ECG data that can be captured in the compressed measurements without losing significant information.

The compressed measurements are fed into the Decoder, which is tasked with reconstructing the original signal from the compressed representation. The decoder consists of a fully connected layer followed by a regression output layer. This part of the model maps the compressed measurements Y back to an approximation of the original signal $\hat{X}_1, \hat{X}_2, ..., \hat{X}_N$. The fully connected layer expands the compressed data back to its original dimension, while the regression layer refines this output, ensuring that the reconstructed signal closely resembles the original ECG segment. By training the encoder and decoder jointly in an end-to-end manner, the model learns an optimal way to compress and reconstruct signals, balancing compression efficiency with reconstruction accuracy. After training, the model was validated on testing dataset and we evaluated the same performance parameters.

5.5 Experimental Results

In this section we discussed the results obtained from above proposed CS frameworks and compare with the state of the art CS sensing matrix and reconstruction algorithms.

5.5.1 Dataset

The proposed method is validated using the MIT-BIH Arrhythmia Database [124,125]. This database includes 30-minute heartbeat recordings from 48 patients, each with two leads. For our experiments, we selected the MLII lead signals. Table 5.1 presents the records used for training and testing datasets. All recordings were segmented into fixed-length segments of N=256 samples, which were then compressed with compression ratios ranging from 10% to 90%.

5.5.2 Quality Evaluations

To evaluate the performance of the proposed method, a set of widely recognized metrics has been employed. Specifically, the quality of reconstruction has been analyzed in relation to the compression capability, expressed as the Compression Ratio (CR). The accuracy of the reconstructed signal has been quantified using the Percent Root-mean-square Difference (PRD) and the corresponding Signal-to-Noise Ratio (SNR), which measure the percentage of error between the original signal vector x and the reconstructed signal vector $\hat{x}$. A comprehensive explanation of these parameters is provided in Chapter 3, Section 3.3.4.

5.5.3 Training Parameters

To calculate the actual difference between the reconstructed signal and the target label, we used the Root Mean Square Error (RMSE) as the loss function to train the network. The loss function can be formulated as follows:

$$L_{\mathrm{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\| f\left(\hat{x}_{(i)}; w, b\right) - x_{(i)}\right\|_2^2} \tag{5.2}$$

where f represents the network operation with weights w and biases b, $x_{(i)}$ is the target signal, $\hat{x}_{(i)}$ is the reconstructed signal, and n denotes the number of signals in the training set.

Adam was chosen as the optimizer for its efficiency and adaptive learning capabilities. The initial training parameters were set as follows:

- **Initial Learning Rate (0.001):** A moderate initial learning rate enables the model to make stable, gradual updates to the weights, improving convergence without large fluctuations.
- **MaxEpochs (200):** The maximum number of epochs was set to 200 to allow sufficient time for convergence while preventing overfitting.
- **L2 Regularization (1e-4):** This regularization parameter helps to prevent overfitting by adding a small penalty to large weights, promoting generalization.
- **MiniBatchSize (300):** A mini-batch size of 300 balances computational efficiency and gradient stability, improving the training speed without sacrificing performance.
- **LearnRateSchedule ('piecewise'):** This schedule reduces the learning rate at specific intervals, allowing finer adjustments as training progresses.
- **LearnRateDropFactor (0.5):** By reducing the learning rate by half, the model can refine its learning as it approaches optimal weights, reducing the chance of overshooting minima.
- **LearnRateDropPeriod (10):** The learning rate decreases every 10 epochs to enable a smoother convergence and adapt the learning rate as the loss function stabilizes.

The experiments were conducted with compression ratios ranging from 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 85% and 90%. The backpropagation algorithm was used to fine-tune all network parameters. The experimental setup included a Windows 11 operating system, MATLAB 2023b, and an Intel(R) Core(TM) i7-13700HX CPU at 2100 MHz with 16 cores and 24 logical processors, ensuring efficient computation and reproducibility.

5.6 Performance Evaluation of the Proposed NN Model for ECG Reconstruction

5.6.1 Performance Evaluation on Data Driven Sensing Matrix

The Data-Driven Sensing Matrices generated at various compression ratios in the previous section were applied to compress the ECG signals. These compressed signals were then reconstructed using the proposed lightweight neural network (NN) model. Figure 5.4 presents a bar plot of the compression ratio (CR) versus the average PRD values on the test data. From the figure, it shows that the proposed NN model achieves a low PRD (Very Good) at lower CR from 10% to 60% with values around 1.3% to 1.94%. Good reconstruction quality at CR 70% and 80% with the values 3.2% to 4.9% and bad quality reconstruction at 85% compression with PRD values higher than 9%. This low PRD up to CR 80% highlights the robustness our proposed NN model in maintaining signal integrity, which is most important for accurate ECG signal diagnostic analysis.

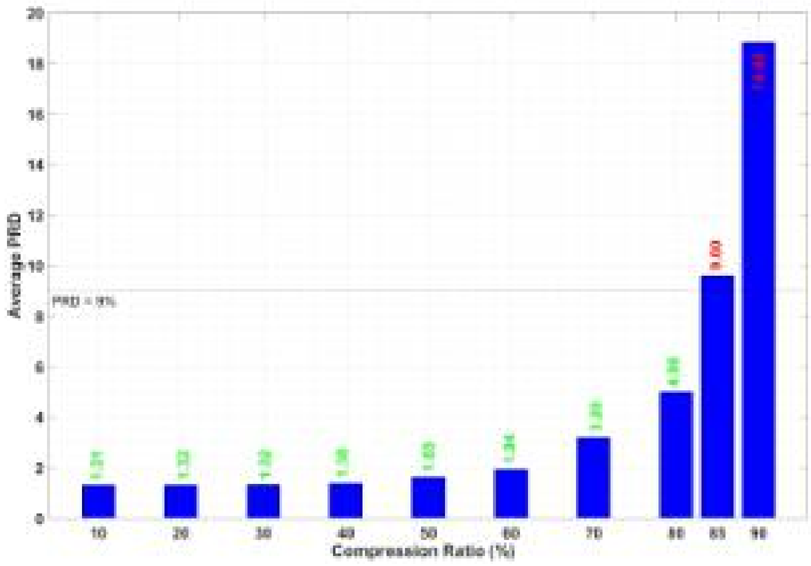

Figure 5.4: Signal Reconstruction Performance of Proposed NN model with DSM

5.6.2 Performance Evaluation on Binary Thresholding Matrix

As discussed in Chapter 4, the data-driven sensing matrices (DSMs) derived from the weights of the encoder layer in the neural network model consist of floating-point values. While these floating-point elements provide high precision, storing them in hardware at the sensing stage can be challenging due to the substantial memory requirements. This increased storage demand impacts resource-constrained devices. To address this limitation, we convert DSMs into binary matrices by applying a thresholding technique, detailed in chapter 4 Algorithm 3. The conversion process involves applying various threshold levels to the DSM, effectively generating Binary Thresholding Matrices (BTMs) with densities ranging from 1% to 15%. By controlling the density of the 1s in the matrix, we aim to find an optimal trade-off between compression efficiency and signal reconstruction quality. A higher density implies more non-zero entries in the matrix, potentially improving reconstruction accuracy but also increasing memory usage. Conversely, a lower density reduces memory requirements but may affect the quality of the reconstructed signal. This balance is critical for practical applications.

We generated the BTMs at various density levels. At each specified density, the ECG data were compressed via the corresponding BTM. After compression, the data is fed into the proposed NN model, which is trained to reconstruct the original ECG signals from the compressed measurements. One NN model was trained, the model is evaluated on a separate test dataset to assess its generalization capabilities across different density levels. The reconstruction quality at each density level is calculated using the PRD.

Figure 5.5 presents the PRD heatmap, illustrating the reconstruction performance across a range of compression ratios (y-axis) and density levels (x-axis). This heatmap provides a visual summary of how PRD varies with both the compression ratio and matrix density, highlighting the trade-offs involved in BTM optimization. Cells in green denote lowest PRD values, indicating optimal combinations of compression ratio and density that yield high reconstruction accuracy.

For instance, 1% density matrices at compression ratios of 10%-40% achieve minimal PRD values of 2.83%, 2.84%, 3.00%, and 3.64%. Similarly, 2% density matrices at compression ratios of 50%-60% achieve low PRD values of 4.03% and 4.53%. At a 70% compression ratio, an 8% density matrix yields a minimum PRD of 7.41%, while a 14% density matrix at an 80% compression ratio results in a PRD of 7.77%. Overall, the results indicate that up to 80% compression with varying density levels can still achieve "good" quality reconstruction (PRD values below 9%), whereas compression ratios above 80% result in "poor" quality reconstruction (PRD values exceeding 9%). The green cells in the heatmap showcase the effectiveness of sparse matrices in capturing essential ECG features with minimal data.

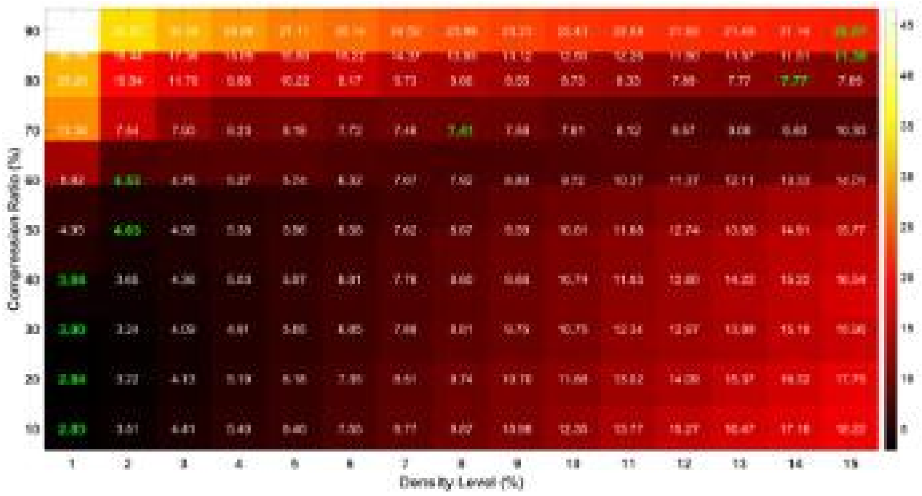

Figure 5.5: PRD Heatmap: Compression Ratio vs Density Level (All PRD Values with Min Highlighted)

The binary thresholding matrix approach enables a more memory-efficient representation of sensing matrices, making it suitable for real-time applications on low-power, memory-constrained devices. By carefully selecting the optimal density and compression ratio, this method provides a flexible, hardware-friendly solution for ECG signal compression and reconstruction, maintaining a balance between efficient data storage and high-quality signal recovery.

5.6.3 Comparison Analysis of DSMs and BTMs

Data-Driven Sensing Matrices (DSMs) and Binary Thresholding Matrices (BTMs) offer two distinct approaches for ECG data compression. DSMs are derived directly from the weights of the encoder layer in the neural network, resulting in matrices with floating-point values that capture data-specific patterns. However, storing and processing floating-point matrices in hardware can be resource-intensive, requiring significant memory and computational power. To address this challenge, BTMs are generated from DSMs by applying a thresholding technique that converts the floating-point values into binary representations. This transformation reduces the memory requirements, making BTMs more suitable for hardware implementation. Figure 5.6 presents the performance comparison between DSMs and BTMs at varying compression ratios. The figure shows the Average PRD on the y-axis, plotted against different Compression Ratios on the

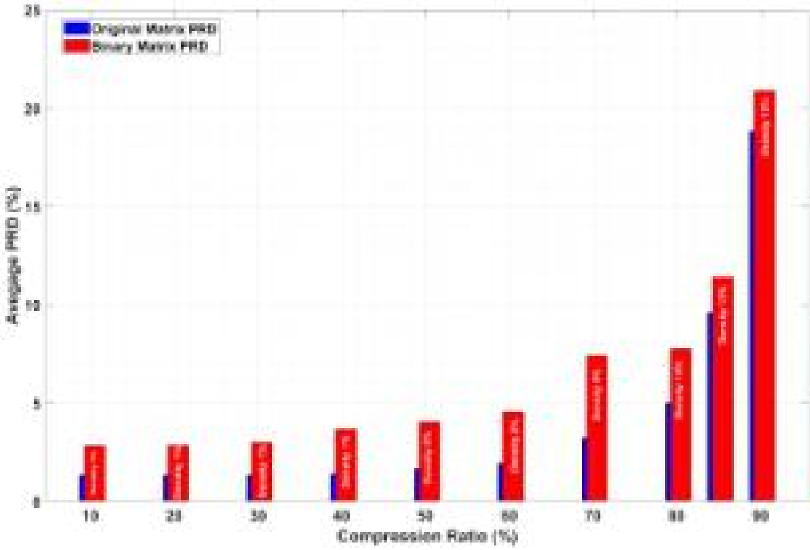

Figure 5.6: Compression ratio vs PRD bar chart for both DSM and BTM sensing matrices

x-axis. The blue bars represent the PRD values for DSMs, while the red bars indicate the PRD values for BTMs at various density levels, labeled within each bar. As shown, DSMs generally achieve lower PRD values compared to BTMs at equivalent compression ratios, indicating that DSMs provide a more accurate reconstruction of the ECG signals. This is expected, as DSMs retain the full precision of the floating-point values learned from the data, which allows for more detailed signal capture. However, BTMs exhibit comparable performance to DSMs at lower compression ratios (10%-50%) and at specific density levels, demonstrating their effectiveness in retaining signal fidelity even with a binary representation. For instance, at a compression ratio of 10%-40% with a 1% density BTM, the PRD remains close to that of the DSM, suggesting that low-density BTMs can still achieve good reconstruction quality while reducing hardware complexity. As the compression ratio increases beyond 60%, the performance gap between DSMs and BTMs becomes more pronounced, particularly at higher densities (e.g., 8%-15%), where the PRD values for BTMs exceed those of DSMs by a significant margin. This indicates that at high compression levels, the binary representation may lose some of the information captured by DSMs, affecting the accuracy of the reconstructed signals.

The comparison results highlight the trade-off between reconstruction accuracy and hardware efficiency. We only optimize the BTM till 15% density, it may achieve less error with higher densities specially at higher compression ratio. The DSMs are ideal for applications requiring high accuracy, BTMs offer a viable alternative for scenarios where hardware efficiency is prioritized. The binary nature of BTMs allows them to be stored and processed with very low memory and computational resources, making them suitable for deployment in low-power devices where resource limitations are a key concern.

5.7 Performance Evaluation of Proposed Autoencoder Model for End-to-End Sensing and Reconstruction

Figure 5.7 presents the relationship between the Compression Ratio (x-axis) and the Average PRD (y-axis), which measures the reconstruction quality at various compression levels. The bars in the figure represent the average PRD achieved by the model at each compression ratio. At compression ratios (10%-70%), the model demonstrates very good reconstruction accuracy, with PRD values consistently below 1.5%. For instance, the PRD at 10% compression is 0.87%, while at 30%, it remains under 1%. These results indicate that even with significant data compression, the model retains high fidelity. As compression ratios increase to 40% to 70% and beyond, the PRD values begin to rise slowly. For example, the PRD is 1.04% at 40% compression and 1.41% at 70%. From CR 80% to 85% it provides good quality reconstruction with PRD value 2.58% at 80% and 7.31% at 85% compression. Although PRD values remain relatively low up to 85% compression levels, the model shows a substantial increase in PRD at compression ratio 90% the PRD is grater than 9% which alternately shows the not good or bad quality reconstruction.

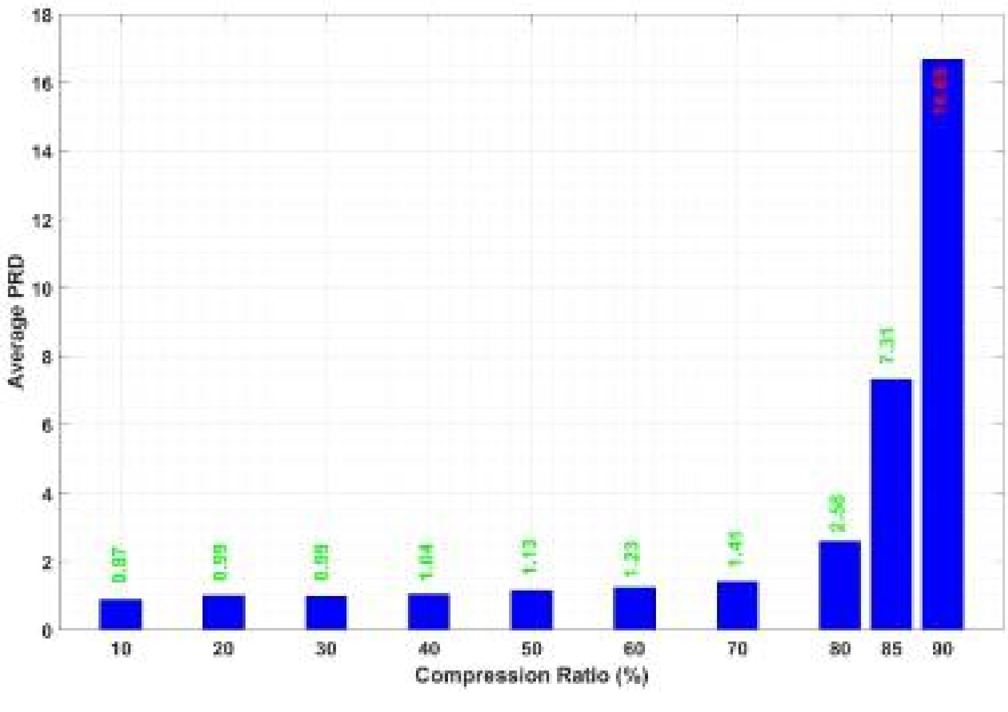

Figure 5.7: Average PRD Values of End-to-End Sensing and Reconstruction

The proposed autoencoder-based end-to-end sensing and reconstruction framework provides an effective solution for ECG Signals. The model achieves very good-quality reconstruction up to 70%, good-quality up to 85% compression, making it ideal for wearable ECG monitoring application and remote health monitoring systems, where both data efficiency and signal accuracy are essential. Additionally we also implemented the encoder and decoder separately on hardware to check their hardware efficiency.

5.8 Comparison with and State-of-the-Art Sensing Matrices and Reconstruction Algorithms

In this section, we present a comprehensive performance comparison of our proposed Data-Driven Sensing Matrix (DSM), Binary Thresholding Matrix (BTM), and Autoencoder-based sensing with traditional fixed sensing matrices, including Random Gaussian Matrix (RGM) and Random Bernoulli Matrices. This comparison is critical to evaluate the effectiveness of our proposed sensing approaches in term of compression ratio, reconstruction quality and signal-to-noise ratio (SNR). Additionally, we compare the performance of our proposed neural network (NN) reconstruction model and autoencoder (end-to-end sensing and reconstruction) model with conventional reconstruction algorithms, such as Basis Pursuit Denoising (BPDN), Orthogonal Matching Pursuit (OMP), and Block Sparse Bayesian Learning (BSBL).

5.8.1 Comparison with Sensing Matrices

Table 5.2 provides a detailed comparison summary at various compression ratios (CR), focusing on two key metrics SPRD (%) and SNR (dB).

The results in Table 5.2 shows the effectiveness of our proposed models in achieving lower PRD and higher SNR across most compression ratios. At a low compression ratio of 10%, the DSM achieves a PRD of 1.312% with an SNR of 39.133 dB, significantly outperforming the RGM (6.673% PRD and 25.063 dB SNR) and BRMB (6.733% PRD and 24.982 dB SNR). The BTM also demonstrates competitive performance with a PRD of 2.827% and an SNR of 32.305 dB, showing its suitability for applications where binary matrices are preferred due to their hardware efficiency. The Autoencoder model achieves the lowest PRD at this compression ratio, with a value of 1.145% and an SNR of 39.719 dB, indicating its superior performance in retaining signal details during compression.

As the compression ratio increases, the proposed models maintain their advantage in reconstruction quality. For instance, at a compression ratio of 50%, the DSM exhibits a PRD of 1.630% and an SNR of 37.090 dB, whereas the RGM and BRMB demonstrate higher PRD values (5.343% and 4.823%, respectively) and lower SNR. The BTM shows an increase in PRD to 4.030%, yet maintains competitive performance in terms of hardware feasibility. The Autoencoder continues to provide excellent reconstruction quality with a PRD of 1.594% and an SNR of 36.830 dB.

At higher compression ratios (70%-90%), the proposed methods still show improved performance compared to traditional sensing matrices, though the PRD values start to increase as expected due to the reduction in retained information. For example, at 90% compression, the DSM achieves a PRD of 18.818% with an SNR of 16.108 dB, while the RGM and BRMB both exhibit PRD values above 30%, indicating significant degradation in reconstruction quality for these traditional matrices. The BTM and Autoencoder models also display elevated PRD values at this compression level, yet they continue to outperform traditional methods.

The overall results demonstrate that our proposed data-driven approaches (DSM, BTM, and Autoencoder) consistently outperform traditional random matrices in both reconstruction accuracy and SNR, especially at higher compression ratios which is our main goal. DSM and Autoencoder model, in particular, shows the highest performance in maintaining signal quality, but at the same time it consume more hardware resources. Furthermore, the BTM presents a hardware-friendly alternative that balances reconstruction quality with reduced memory and computational demands, essential for real-time applications on resource-limited devices.

Table 5.2: PRD and SNR Comparison Table

CR	RGM		BRMS		DSM (Proposed)		BTM (Proposed)		Autoencoder (Proposed)	
	PRD [%]	SNR	PRD [%]	SNR	PRD [%]	SNR	PRD [%]	SNR	PRD [%]	SNR
10	6.673	25.063	6.733	24.982	1.312	39.133	2.827	32.305	1.145	39.719
20	7.034	24.581	6.869	24.793	1.318	39.096	2.838	32.319	1.292	38.527
30	6.522	25.216	6.590	25.135	1.320	39.079	2.996	31.811	1.337	38.376
40	5.521	26.651	5.823	26.167	1.382	38.669	3.638	30.204	1.452	37.692
50	5.343	26.871	4.823	27.757	1.630	37.090	4.030	29.219	1.594	36.830
60	4.182	28.881	4.130	29.013	1.936	35.597	4.531	28.272	1.865	35.573
70	5.349	26.649	5.255	26.797	3.198	31.217	7.408	24.069	2.369	33.589
80	11.286	20.199	11.444	20.085	4.986	27.488	7.767	23.651	4.195	28.902
85	19.447	15.470	19.216	15.561	9.605	22.028	11.389	20.520	9.103	22.499
90	31.548	11.145	32.885	10.735	18.818	16.108	20.874	15.078	18.832	16.087

5.8.2 Comparison with traditional Reconstruction Algorithms

In this section, we conduct a comprehensive comparison between the proposed reconstruction methods and several other CS techniques, including both traditional and deep learning-based approaches. Specifically, we compare our methods with three conventional CS reconstruction techniques (BPDN [149], OMP [174], and BSBL-BO [175]. Firstly, we assess the reconstruction accuracy of each method using two commonly employed metrics PRD and SNR. Secondly, we evaluate the reconstruction time for each method to understand the computational efficiency of the proposed approach in comparison with traditional CS methods. Reconstruction time is a critical factor, especially in real-time applications, where low-latency processing is essential for continuous ECG monitoring. Lastly, we examine the quality of the reconstructed QRS complexes, which are critical features in ECG signals. This aspect of the comparison provides insight into how well each method retains diagnostically relevant features in the reconstructed ECG signals. Among the traditional methods, BPDN is based on convex optimization techniques, OMP relies on a greedy algorithm, and BSBL-BO employs a sparse Bayesian learning framework. For BPDN and OMP, we use the rbio5.5 wavelet as the sparse basis, as this basis has demonstrated effectiveness in representing ECG signals sparsely. Additionally, a random Gaussian matrix is used as the measurement matrix for both BPDN and OMP, while a sparse random matrix is utilized in BSBL-BO.

5.8.3 Evaluation of Reconstruction Error

The average PRD and SNR values across different CRs for each method on the test set are shown in Figure 5.8(a) and (b), respectively. Figure 5.8(a) shows that at high compression ratios (CR $\geq$ 70%), the proposed models achieve significantly lower PRD values than traditional reconstruction methods, indicating a lower reconstruction error. Additionally, the PRD growth in our models remains relatively stable, even as CR increases, which suggests that the proposed methods are more robust at maintaining reconstruction quality across a wide range of compression levels. Similarly, Figure 5.8(b) indicate that both the DSM-NN and Autoencoder models outperform traditional methods in terms of average SNR at all tested compression ratios. This means that these models retain a higher signal quality during reconstruction. The BTM-NN model also shows superior performance at higher CRs (CR $\geq$ 80%), highlighting its effectiveness in scenarios that require more aggressive data compression. To further illustrate the

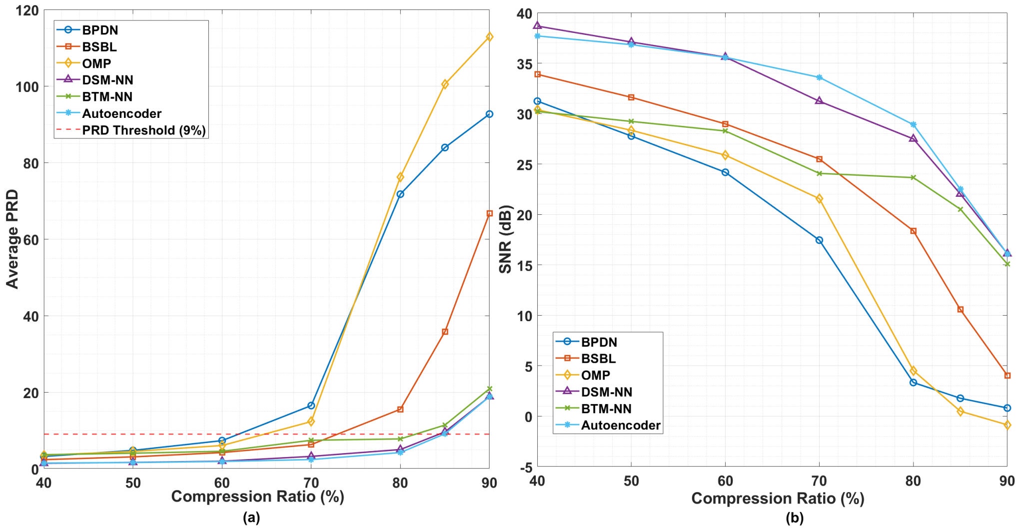

Figure 5.8: Averaged PRD and SNR Comparison at different CRs

performance of each method, Figure 5.9 provides a visual comparison between the original ECG signals and the reconstructed signals at a compression ratio of 85% for approximately 3 seconds of data from record 121 (first row) and record 210 (second row). In these visualizations, it is evident that the proposed models, particularly DSM-NN, BTM-NN, and Autoencoder, are able to preserve the QRS complex, the most important feature for diagnosing cardiac events—while traditional methods like BPDN, OMP, and BSBL-BO struggle to maintain signal quality. The poor reconstruction quality in these traditional methods results in significant loss of diagnostic information, as reflected in the distorted QRS complexes.

5.8.4 Reconstruction Time Analysis

The reconstruction time is critical in CS for ECG signals, especially for real-time healthcare monitoring systems. Table 5.3 presents the average execution time required for reconstructing one ECG segment of 256 samples using various algorithms across different compression ratios (CRs). Our proposed methods, DSM-NN, BTM-NN, and Autoencoder, exhibit remarkable speed advantages compared to traditional reconstruction algorithms. Across different CRs, the reconstruction time for DSM-NN, BTM-NN, and Autoencoder is significantly reduced compared to BPDN, OMP, and BSBL-BO. The substantial reduction in reconstruction time achieved by our models can be attributed to their non-iterative architecture, which facilitates effective parallel processing. By minimizing iterative calculations, our methods not only enhance computational efficiency but also make the proposed models highly suitable for hardware implementation.

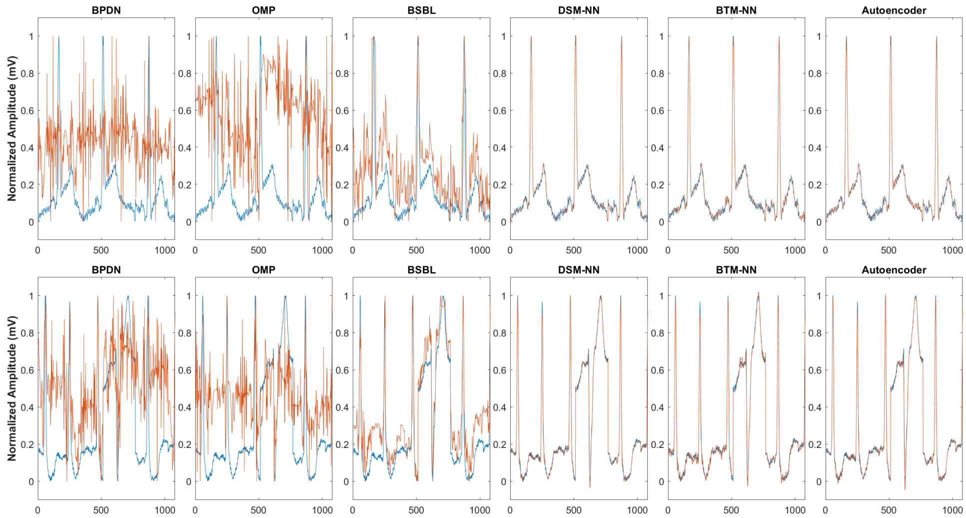

Figure 5.9: Record 121 and record 210 original and reconstructed signals over approximately 3s, when CR = 85%. Row 1 belongings to record 121 and row 2 belongings to record 210, while columns belongs to algorithms

Table 5.3: Reconstruction Time (in milliseconds) of Various Algorithms for Reconstructing a Segment of Length 256 Samples

Algorithms	Reconstruction Time (ms)						
	CR-40	**CR-50**	**CR-60**	**CR-70**	**CR-80**	**CR-85**	**CR-90**
BPDN	506.3558	476.7339	453.2163	1032.6611	427.0555	381.9328	349.0024
BSBL	52.1712	45.6413	40.6713	37.1179	32.7410	31.0414	30.1742
OMP	19.8932	18.3425	11.8869	7.1419	3.7448	2.2804	1.2126
DSM-NN (Proposed)	1.2574	1.2492	1.2320	1.2171	1.1931	1.1609	1.1127
BTM-NN (Proposed)	1.2898	1.2831	1.2733	1.2120	1.2095	1.2009	1.9212
Autoencoder (Proposed)	0.4906	0.4916	0.4907	0.2863	0.2723	0.2563	0.2157

5.8.5 Hardware Implementation

To extend the real-time application of our proposed CS framworks for ECG signal, we implemented on STM32F401 Nucleo microcontroller, which operates at 84 MHz clock frequency with 512 KB of Flash memory and 96 KB of SRAM. We measured the processing inference time required to compress and reconstruct one complete segment of ECG data. Additionally we measured the power and energy consumption during compression and reconstruction.

5.8.6 Hardware implementation of Binary Thresholding Sensing Matrix

Table 5.4 provides the summary of hardware resources across various compression ratios, including the processing time, power consumption, and energy usage. Figure 5.10 shows the timing diagram of micorocontroller which processes ECG signals segment with an original sampling frequency of 360 Hz, meaning that a new sample arrives approximately every 2.78 milliseconds or 2780 microseconds. Given the processing times observed, which range from 61 to 60 microseconds depending on the compression ratio, the BTM implementation easily fits within this time window. In other words, the time required to compress each segment is significantly shorter than the interval between incoming samples, confirming that this implementation is well-suited for real-time applications.

Table 5.4: Microcontroller Implementation Results

Compression Ratio	Density	Processing Time (microseconds)	Clockcycles	Power (mW)	Energy (µJ)
10	1.0	24	2026	71.28	1.71072
20	1.0	21	1757	71.28	1.49688
30	1.0	19	1558	71.28	1.35432
40	1.0	16	1370	71.28	1.14048
50	2.0	25	2087	71.28	1.782
60	2.0	21	1707	71.28	1.49688
70	8.0	52	4402	71.28	3.70656
80	14.0	60	5061	71.28	4.2768

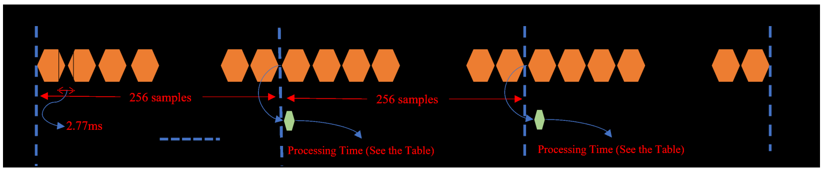

Figure 5.10: Timing diagram of Microcontroller Implementation

Additionally, We estimated the current consumption using the PCC Power Consumption Calculator in STM32CubeMX and subsequently calculated the power and energy requirements based on the processing time for each compression ratio as shown in table 5.4.

5.8.7 Hardware Implementation of Compression and Reconstruction Models

This section presents the results of deploying a ECG compression and reconstruction framework on the STM32 NUCLEO-F401RE microcontroller. As discussed in the section 5.7 and 5.4.3, the encoder model performs the task of signal compression, while the decoder model reconstructs the compressed signal. Additionally, the same reconstruction model is used for reconstructing signals from compressed measurements with binary thresholding sensing matrix and a data-driven sensing matrix. All models were initially trained in MATLAB, and these trained models were subsequently deployed on the microcontroller via the ST Edge Cloud AI platform [176].

The results in Tables 5.5 and 5.6 compile the resource requirements for both the encoder (compression) and decoder (reconstruction) models across various compression ratios. Key metrics, including clock cycles, inference time, model size, RAM usage, and flash memory usage, provide insights into the computational and memory efficiency achieved at each compression ratio.

- **Resource Consumption for Compression Model (Table 5.5):** The results indicate a notable reduction in computational and memory demands as the compression ratio increases from 10% to 90%. This reduction is particularly evident in the clock cycles and inference time required for the compression model.

  - **Clock Cycles and Inference Time:** At a low compression ratio (10%), the model requires a substantial 444.2k clock cycles and 5.288 ms inference time. As the compression ratio increases to 90%, these values decrease significantly to 48.8k clock cycles and 0.581 ms inference time. This pattern demonstrates that higher compression ratios reduce the computational load, as the model processes a smaller volume of data, thereby accelerating the compression operation. According to the timing diagram in Figure 5.10, a new sample is received every 2.77 ms. At compression ratios from 10% to 50%, the inference or elaboration time exceeds 2.77 ms, making real-time implementation infeasible at a sampling frequency of 360 Hz on this microcontroller. However, with compression ratios from 60% to 90%, the inference time is below 2.77 ms, allowing real-time implementation to be achieved at these compression levels. It is also worth noting that high-quality reconstruction is maintained up to a compression ratio of 85%, as shown in Table 5.2.
  - **Memory Utilization:** Memory requirements, as indicated by model size, RAM, and flash memory usage, also diminish with increasing compression ratios. For instance, at a 10% compression ratio, the model size stands at 236,440 bytes, but it is reduced to 25,700 bytes at 90% compression. This trend underscores the memory efficiency achievable through greater compression, which is critical for applications with stringent memory constraints.

- **Resource Consumption for Reconstruction Model (Table 5.6):** The decoder model, responsible for reconstructing the compressed signal, similarly benefits from increased compression ratios, with improvements in both computational speed and memory efficiency.

  - **Clock Cycles and Inference Time:** The reconstruction model displays a reduction in clock cycles and inference time as compression ratios increase. For example, at a 10% compression ratio, the model requires 443,300 clock cycles and 5.277 ms to perform reconstruction, whereas at a 90% compression ratio, these values decrease to 59,300 clock cycles and 0.706 ms, respectively. This suggests that higher compression ratios facilitate a faster signal reconstruction process, a desirable feature for time-sensitive applications.
  - **Memory Utilization:** Memory demand decreases in parallel with the increased compression ratio. The model size at a 10% compression ratio is 236,544 bytes, but it declines to 26,624 bytes at a 90% compression ratio, highlighting the efficient memory use at higher compression levels. Such memory savings are essential for optimizing embedded systems where resources are limited.

The trends across both models underscore the benefits of higher compression ratios in reducing resource consumption. As compression ratios increase, both the computational load and memory requirements decrease, positioning high compression as a viable approach for embedded applications that demand efficient use of limited resources. However, increased compression may impact the fidelity of the reconstructed signal, a factor that must be balanced against the gains in computational efficiency.

Table 5.5: Compression Model Implementation on Microcontroller

Microcontroller: STM32 NUCLEO-F401RE, Clock Frequency: 84MHz					
Compression Ratio (%)	**Clock Cycles (x10^3)**	**Inference Time (ms)**	**Model Size (Bytes)**	**RAM (Bytes)**	**Flash Memory (Bytes)**
10	444.2	5.288	236440	2936	244278
20	394.0	4.690	209712	2832	217550
30	345.8	4.116	184012	2732	191850
40	295.6	3.519	157284	2628	165122
50	247.4	2.946	131584	2528	139422
60	197.3	2.349	104856	2424	112694
70	147.2	1.752	78128	2320	85966
80	98.9	1.178	52428	2220	60266
85	73.9	0.880	39064	2168	46902
90	48.8	0.581	25700	2116	33538

Table 5.6: Reconstruction Model Resource Consumption on Microcontroller

Microcontroller Specifications: STM32 NUCLEO-F401RE, Clock Frequency: 84MHz					
Compression Ratio	**Total Clock Cycles (x10^3)**	**Inference Time (ms)**	**Model Size**	**RAM Memory (Bytes)**	**Flash Memory (Bytes)**
10	443.3	5.277	236544	2764	243634
20	394.6	4.698	209920	2660	217010
30	346.1	4.120	184320	2560	191410
40	299.3	3.563	157696	2456	164786
50	250.7	2.985	132096	2356	139186
60	203.6	2.423	105472	2252	112562
70	154.9	1.844	78848	2148	85938
80	106.4	1.266	53248	2048	60338
85	83.7	0.997	39936	1996	47026
90	59.3	0.706	26624	1944	33714

5.9 Discussion

The results from the deep learning-based compressed sensing (CS) framework presented in this chapter demonstrate the efficacy of data-driven and adaptive approaches in handling ECG signal compression and reconstruction. Traditional CS techniques, though effective, often encounter limitations in resource-constrained environments due to fixed measurement matrices and the need for signal sparsity transformations. The proposed models overcome these challenges through light weight Neural Network architecture that incorporates end-to-end training and learnable measurement matrices designed for ECG signals.

The data-driven sensing matrix (DSM) and binary thresholding matrix (BTM) offer two distinct yet complementary approaches for optimizing memory and processing efficiency. The DSM, learned directly from ECG signal features, provides high reconstruction accuracy across a wide range of compression ratios but requires significant computational resources. Conversely, the BTM provides a more hardware-efficient solution by converting DSM elements into binary representations, making it suitable for low-power and real-time applications. The autoencoder-based model further enhances the adaptability of the CS framework by jointly optimizing both sensing and reconstruction processes, achieving high reconstruction quality up to an 85% compression ratio. This integration makes it a reliable solution for wearable devices and telemedicine applications where both accuracy and computational efficiency are important.

The experimental results confirm that the proposed CS framework and NN models such as BTM and autoencoder models, consistently outperform traditional sensing matrices in terms of PRD, SNR and reconstruction time. Furthermore, as compression ratios increase, the memory and computational load reduce substantially, which is important for real time and embedded applications. It is worth noting that very high compression levels (above 85%) the traditional CS methods compromise reconstruction fidelity, but our proposed models achieve good quality reconstruction at hight compression ratios.

The deployment of these models on the STM32 NUCLEO-F401RE microcontroller validates their feasibility for real time applications. Resource consumption metrics, including clock cycles, inference time, and memory usage, suggest that higher compression ratios enable real-time performance while reducing memory demands—a key advantage for edge computing applications in health monitoring.

5.10 Future Work

The promising results obtained in this study open some challenges for future research and improvements. One potential direction is to enhance the robustness of the proposed models under varying physiological and environmental conditions. Incorporating diverse ECG datasets that encompass a wide range of heart conditions, age groups, and noise artifacts would improve the generalization of the models. Furthermore, while this study focused on the MIT-BIH Arrhythmia Database, future work could explore additional datasets to assess the adaptability of the proposed models to different signal patterns and noise levels encountered in real-world monitoring.

Another avenue for research lies in optimizing the data-driven sensing matrix (DSM) and binary thresholding matrix (BTM) for higher compression ratios while minimizing reconstruction error. Advanced deep learning architectures, such as recurrent neural networks (RNNs) or transformers, could be explored to capture temporal dependencies in ECG signals more effectively. In this study we only implemented on STM32F401 Nucleo board, In future, expanding the deployment to other wearable platforms beyond the STM32 NUCLEO-F401RE microcontroller could increase the practical applicability of these models in various healthcare and telemedicine contexts.

The limitation is the reliance on high compression ratios, which, while reducing resource requirements and impact reconstruction quality. Although the models achieve good reconstruction quality up to an 85% compression ratio, compression beyond this level often compromises signal fidelity, posing challenges for accurate diagnostics. Additionally, the hardware implementation of the binary thresholding matrix (BTM) introduces memory savings, yet it reduces reconstruction quality at high compression ratios compared to the floating-point data-driven sensing matrix (DSM). This trade-off between hardware efficiency and signal fidelity should be considered in applications where diagnostic precision is important. Finally, while the STM32 NUCLEO-F401RE microcontroller serves as an effective platform for testing, its specific architecture may limit the portability of these models to other embedded devices without further hardware adaptations.

5.11 Conclusion

In this chapter, we introduced a light weight Neural Network framework for compressive sensing of ECG signals, employing a data-driven approach to enhance compression efficiency and reconstruction quality. By implementing end-to-end sensing and reconstruction models, we addressed limitations inherent to traditional CS methods, such as the dependency on fixed measurement matrices and signal sparsity. The proposed data-driven sensing matrix (DSM) and binary thresholding matrix (BTM) models offer robust solutions for hardware-friendly compression and high-quality signal reconstruction, with the autoencoder model excelling in end-to-end optimization of compression and reconstruction processes.

The DSM and autoencoder models achieve high-quality ECG signal reconstruction at compression ratios up to 85%, indicating their potential for accurate cardiac monitoring on resource-constrained platforms. The BTM model, with its binary structure, provides a scalable solution for low-power devices, making it especially suitable for real-time applications. Experimental results on the STM32 NUCLEO-F401RE microcontroller demonstrated that these models can achieve real-time performance with minimal memory and energy consumption, thereby supporting their deployment in wearable ECG monitoring systems.

This work contributes to the development of efficient, adaptive, and scalable solutions for ECG signals, advancing the capabilities of deep learning-based CS methods in healthcare applications. Future research may further optimize these models for broader biomedical applications and explore the potential of hybrid sensing matrices to balance computational efficiency with reconstruction fidelity, expanding the practical applicability of deep learning in compressed sensing.

Chapter 6

Learning on Compressed ECG Measurements

6.1 Introduction

Electrocardiograms (ECG) are among the most widely used physiological signals in healthcare, playing a vital role in assessing heart health and detecting potential abnormalities. For clinicians and researchers, ECGs offer invaluable insights into cardiac function, making them crucial tools for diagnosing a range of conditions, from arrhythmias to heart failure. However, ECG signals are often high-dimensional, presenting unique challenges in terms of data storage, transmission, and processing, particularly in settings where resources are limited.

Traditionally, ECG signals are sampled at high rates, following the Nyquist-Shannon theorem to capture all necessary details. While effective, this approach generates large volumes of data that demand substantial storage and computational resources. In response to these challenges, researchers have turned to compressed sensing (CS) as an innovative solution. Compressed sensing capitalizes on the inherent sparsity of ECG signals, allowing for accurate reconstruction from a compressed measurements. This process significantly reduces data size while preserving the critical diagnostic information required for clinical assessments.

Despite its potential, compressed sensing often involves complex and computationally intensive reconstruction steps, which can be impractical for real-time applications. This limitation has led to the development of compressed learning (CL) approach that bypasses full reconstruction. Instead, CL enables direct feature extraction from compressed measurements, streamlining the process considerably. Recent studies have shown that compressed learning can be highly effective for ECG analysis, demonstrating strong results in detecting abnormalities such as atrial fibrillation (AF) using machine learning models trained directly on compressed data.

However, despite these promising advances, existing CL methods encounter significant challenges, particularly when applied at high compression ratios. Key issues include difficulties in optimizing feature extraction directly from compressed data, as well as limited use of traditional classifiers, which may impact diagnostic accuracy. This chapter addresses these challenges by proposing an optimized compressed learning method. Our approach focuses on extracting clinically relevant features such as cardiac arrhythmia (ARR), congestive heart failure (CHF), and normal sinus rhythm (NSR) directly from compressed ECG signals. Furthermore, we evaluate the effectiveness of this approach by comparing it with traditional classifiers, including SVM, K-NN, and neural network (NN), offering a thorough assessment of their strengths and limitations in handling compressed data for ECG classification.

The primary contributions of this thesis include:

- Proposing a compressed learning approach for efficient ECG classification that reduces computational complexity by enabling direct feature extraction from compressed measurements.
- Developing and evaluating a deep learning-based classification model specifically employed for compressed ECG data, aiming to improve classification accuracy for ARR, CHF, and NSR.
- Evaluate the model on signal compressed from four different sensing matrices.
- Conducting a comparative analysis between conventional machine learning techniques (e.g., SVM, KNN and NN) and the proposed CNN model for ECG classification.

6.2 Related Work and Motivation

6.2.1 Background

Physiological signals, such as electrocardiograms (ECG), are critical for assessing cardiac health, yet their high-dimensional nature creates significant storage and transmission challenges, particularly in wearable and remote health monitoring devices. Traditional sampling techniques leading to large data volumes that are not always practical for continuous monitoring and storage [177]. Compressed sensing (CS) has emerged as a promising solution for reducing data dimensionality by enabling sparse signal reconstruction from fewer measurements, making it highly relevant for ECG data transmission and storage in low-power applications [94]. However, the reconstruction step in CS remains computationally expensive, often prohibiting its use in real-time scenarios. Compressed Learning (CL) addresses this limitation by directly extracting features from compressed measurements, thus bypassing the reconstruction step and reducing computational complexity [178].

6.2.2 Related Work

Several studies have contributed to the application of compressed sensing and compressed learning for ECG signal processing. For instance, Da Poian et al. [179] proposed Compressed Sensing Matched Filtering (CSMF) for detecting R-peaks in compressed ECG data, which enables the extraction of RR intervals without full signal reconstruction. Building on the concept of CL, Zhang et al. [180] introduced a simple deterministic measurement matrix (SDMM) and a Transposed Projection-Convolutional Neural Network (TP-CNN) to enhance AF detection in wearable devices, achieving high accuracy with reduced power consumption. Similarly, Siddique et al. [181] explored the use of low-power approximate adders in digital CS for IoT healthcare systems, illustrating that critical event detection in biomedical signals can be achieved with energy efficiency.

Further advancements in compressed ECG analysis include the use of specific machine learning models. Abdelazez et al. [178] employed Random Forest on compressed ECG data to detect atrial fibrillation (AF) using discrete cosine transform, statistical methods, and wavelet transforms. This approach demonstrated reliable AF detection, affirming the feasibility of machine learning in compressed signal classification. Additionally, a study by Laudato et al. [182] introduced a method for heart rate estimation by identifying R-peaks in a single-lead ECG using a circulant sensing matrix and random forest classifier, contributing valuable insights into low-resource, high-efficiency classification methods for compressed data. Another study by Zhang et al. [183] utilized a 1-D Convolutional Neural Network for feature extraction directly from compressed ECG signals, further underscoring the potential of deep learning in the CL domain.

In the field of ECG classification from compressed data, some studies have explored the potential of structured and learned matrices to optimize compressed measurements. For example, the Structured Fourier Matrix (SFM) has shown promise for faster acquisition and lower storage requirements compared to traditional random Gaussian matrices [184]. Meanwhile, Kumar et al. [76] reviewed the importance of choosing the correct measurement matrix, particularly highlighting the advantages of structured matrices and the benefits of learned matrices, such as neural network-based approaches, in providing adaptive and effective compression.

6.2.3 Motivation

The growing demand for real-time cardiac monitoring in resource-limited settings, particularly in wearable health devices, necessitates the development of efficient and low-power data processing techniques. Existing methods, while promising, often fall short in handling high compression ratios effectively, which can impact classification accuracy. Furthermore, traditional approaches rely on computationally expensive reconstruction steps, which are impractical for real-time applications. This research is motivated by the potential of compressed learning to address these limitations, providing a framework that enables real-time ECG classification without requiring full signal reconstruction. By exploring the use of both traditional machine learning classifiers (e.g., SVM, KNN, NN) and deep learning models (e.g., CNN), this chapter aims to establish an efficient solution for ECG signal classification directly from compressed data, thereby enhancing the feasibility of real-time, low-power cardiac monitoring systems.

6.3 Compressed Learning for ECG Signal Classification

Compressed Learning (CL) extends the concept of compressed sensing by allowing direct feature extraction from compressed data, which reduces computational complexity and energy consumption in real-time applications. Mathematically, the compressed ECG signal y can be represented as:

$$y = \Phi \cdot x \tag{6.1}$$

where $x \in \mathbb{R}^{N\times 1}$ is the original ECG signal, $\Phi \in \mathbb{R}^{M\times N}(M \ll N)$ is the measurement matrix, and $y \in \mathbb{R}^{M\times 1}$ is the compressed signal. Traditional CS approaches require reconstruction of x from y, which can be computationally intensive. In contrast, CL methods bypass reconstruction, directly extracting features from y for classification purposes.

The choice of measurement matrix Φ is crucial in CL, as it determines the quality of compressed measurements and the accuracy of feature extraction. In this chapter, we explored various matrices, including random Gaussian matrices, random binary matrix structured Fourier matrices, and block diagonal matrix, to optimize compressed feature representation for ECG classification.

Using the compressed ECG features, different machine learning classifiers such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Neural Networks (NN) are evaluated, with a focus on achieving high classification accuracy for arrhythmia (ARR), congestive heart failure (CHF), and normal sinus rhythm (NSR). Additionally, we proposed a Convolutional Neural Network (CNN) based deep learning model and trained directly on compressed measurements, to enhance feature extraction and classification accuracy. Figure 6.1 illustrates the compressed learning process for ECG signal classification, where the feature extraction occurs directly on the compressed data, eliminating the need for signal reconstruction.

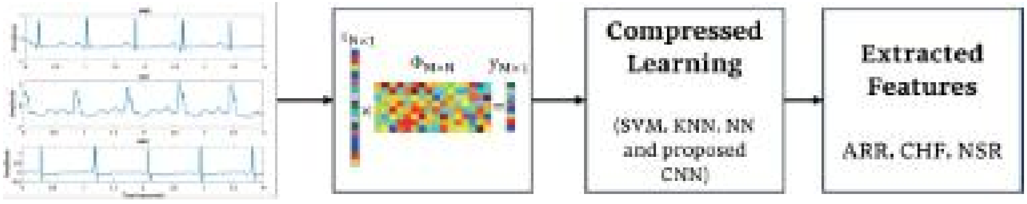

Figure 6.1: Compressed Learning Framework for ECG Signal Classification. The framework enables direct feature extraction and classification of ECG signals in the compressed domain, reducing computational overhead.

6.4 Methods and Materials

This section details the methodology and materials used in this study for developing and evaluating a Compressed Learning (CL) framework for ECG signal classification. The proposed framework integrates various sensing matrices and a proposed classification model. Furthermore, we discuss the databases employed, as well as the performance evaluation metrics that were applied to assess model effectiveness across various settings.

6.4.1 Dataset

For this study, we utilized three publicly accessible ECG datasets from PhysioNet: the BIDMC Congestive Heart Failure Database, the MIT-BIH Normal Sinus Rhythm Database, and the MIT-BIH Arrhythmia Database. Each of these databases was originally sampled at different frequencies, so to maintain consistency across our classification tasks, we downsampled the BIDMC Congestive Heart Failure Database and the MIT-BIH Arrhythmia Database to 128 Hz to match the MIT-BIH Normal Sinus Rhythm Database. Additionally, we divided each dataset into training and testing sets, as outlined in Table 6.1. Below, we describe each database and its contribution to our study.

Table 6.1: Databases Training and Testing Set Distribution

Training Set (80%)	Testing Set (20%)
Database: BIDMC Congestive Heart Failure (Total Records: 15)	
Training Set (80%)	**Testing Set (20%)**
chf01, chf02, chf03, chf04, chf05, chf06, chf07, chf08, chf09, chf10, chf11, chf12	chf13, chf14, chf15
Database: MIT-BIH Normal Sinus Rhythm (Total Records: 18)	
Training Set (80%)	**Testing Set (20%)**
16265, 16272, 16273, 16420, 16483, 16539, 16773, 16786, 16795, 17052, 17453, 18177, 18184, 19088, 19090	19093, 19140, 19830
Database: MIT-BIH Arrhythmia Database (Total Records: 47)	
Training Set (80%)	**Testing Set (20%)**
100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 117, 118, 119, 121, 122, 123, 124, 200, 201, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 217, 219, 220	221, 222, 223, 228, 230, 231, 232, 233, 234

MIT-BIH Normal Sinus Rhythm Database

The MIT-BIH Normal Sinus Rhythm Database [124] includes 18 long-term ECG recordings from healthy individuals without significant arrhythmias. This dataset, sampled natively at 128 Hz, contains recordings from 5 men (aged 26 to 45) and 13 women (aged 20 to 50). As indicated in Table 6.1, we allocated 15 records (80%) for training and 3 records (20%) for testing. This dataset provides a baseline for normal cardiac rhythms and enables the model to distinguish between healthy and pathological signals.

BIDMC Congestive Heart Failure Database

The BIDMC Congestive Heart Failure Database [185] contains long-term ECG recordings from 15 individuals diagnosed with severe congestive heart failure (NYHA class 3–4), including 11 men (aged 22 to 71) and 4 women (aged 54 to 63). Each recording spans approximately 20 hours, recorded over two ECG channels at an original sampling frequency of 250 Hz. For this study, we downsampled these signals to 128 Hz, ensuring compatibility with the other datasets. As shown in Table 6.1, 12 records (80%) were selected for training, and the remaining 3 records (20%) were used for testing.

MIT-BIH Arrhythmia Database

The MIT-BIH Arrhythmia Database [124, 125], a widely used dataset in arrhythmia research, contains 48 half-hour ECG recordings from 47 individuals, sampled at 360 Hz. This dataset captures a variety of arrhythmic events, making it invaluable for arrhythmia detection. To align with the other datasets, we downsampled this database to 128 Hz. From the 47 records, we used 39 (80%) for training and 9 (20%) for testing, as summarized in Table 6.1.

6.4.2 Proposed CL Framework for ECG Classification

The proposed Compressed Learning (CL) framework for ECG classification is designed to classify ECG signals into three categories: Arrhythmia (ARR), Congestive Heart Failure (CHF), and Normal Sinus Rhythm (NSR). The overall framework, as illustrated in Figure 6.2, comprises multiple stages, including data acquisition, signal compression, and classification.

The input ECG signals are obtained from three distinct datasets containing ARR, CHF, and NSR signals (section 6.1). As shown in the left section of Figure 6.2, examples of these signal types are presented to illustrate the variations in amplitude and waveform shape across different conditions. Each signal undergoes segmentation into smaller samples to allow efficient processing and analysis. For this study, various segment sizes (128, 256, 512, 1024, and 2048 samples) were tested, with the segment size of 128 yielding the best performance in terms of classification accuracy and computational efficiency.

The next step in the framework involves compressing the ECG segments using a sensing matrix, shown in the middle section of Figure 6.2. The sensing matrix ($\Phi_{M\times N}$) is designed to take the original signal ($X_{N\times 1}$) and transform it into a compressed version ($Y_{M\times 1}$). We explored four different types of sensing matrices named Random Gaussian, Random Binary, Structured Fourier, and Block Diagonal sensing matrices. Each bringing unique properties to the compressed measurements. By adjusting the compression ratio (ranging from 10% to 90%), we evaluated how well the model performed under various levels of data reduction.

The compressed signals are then fed into a CNN, which serves as the core of the proposed classification model. As illustrated in the right section of Figure 6.2, the CNN consists of multiple layers, each designed to capture essential features from the compressed ECG signals. The network begins with a Conv1D layer, followed by batch normalization and max pooling layers. This combination helps the model learn hierarchical patterns in the data, with each convolutional layer capturing increasingly complex features. Additional Conv1D layers, batch normalization, and max pooling layers further refine the feature extraction process, ultimately producing a rich set of features for classification.

The flattened feature map is then passed through two fully connected (FC) layers (denoted as FC1 and FC2), which further process the extracted features. These fully connected layers play a important role and enabling the model to classify the signals accurately into one of the three categories (ARR, CHF, or NSR). The final output layer produces the classification results, indicating the predicted category of each input ECG segment.

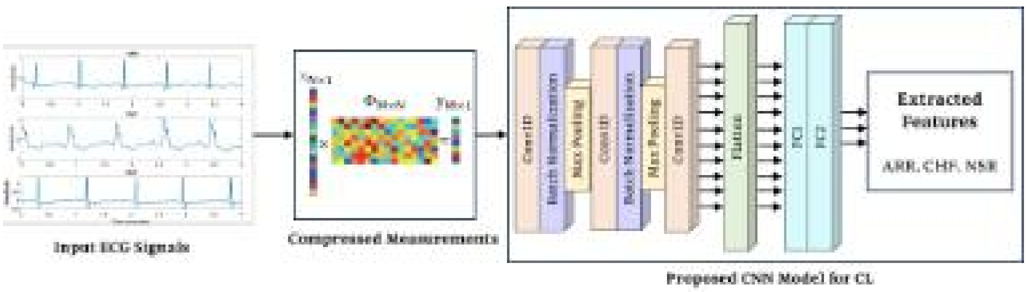

Figure 6.2: Proposed CNN-based CL framework for ECG signal classification

Table 6.2: Proposed CL Model Parameters at CR=50%

No.	Layer Name	Kernel Size	Filters	Padding	Stride	Output Shape	Parameters
1	Conv1D	2	64	0	1	(64, 64)	192
2	MaxPooling1D	-	-	0	2	(32, 64)	0
3	BatchNormalization	-	-	-	-	(32, 64)	256
4	Conv1D	2	128	0	1	(32, 128)	16512
5	MaxPooling1D	-	-	0	2	(16, 128)	0
6	BatchNormalization	-	-	-	-	(16, 128)	512
7	Conv1D	1	256	0	1	(16, 256)	33024
8	Flatten	-	-	-	-	4096	0
9	Dense	-	-	-	-	128	524416
11	Dense	-	-	-	-	64	8256
13	Dense	-	-	-	-	3	195

Table 6.2 summarizes the architecture of this CNN model at compression ratio (CR) of 50%. Each layer in the network serves a unique role in feature extraction and classification, with convolutional layers capturing spatial patterns and dense layers facilitating classification decisions. The network's performance across various compression ratios is analyzed in terms of classification accuracy, F1 score and recall providing insights into the model's robustness in handling different compression levels.

6.4.3 Performance Evaluation Metrics

We employ a range of performance metrics to provide a view of the model performance. Including accuracy, precision, recall, F1-score, computational efficiency, and compression ratio. Each metric provides unique insights into different aspects.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.2}$$

Accuracy is the proportion of correctly classified samples out of the total samples. Accuracy gives us an overall measure of how well the model performs across all classes.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.3}$$

Precision or positive predictive value, represents the proportion of true positive predictions out of all positive predictions. High precision indicates that the model is reliable in its positive classifications, reducing false positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6.4}$$

Recall also known as sensitivity, measures the model's ability to identify all true instances of a particular class. High recall minimizes false negatives, which is specifically important in medical applications where missed detections can have serious implications.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.5}$$

F1-Score is the harmonic mean of precision and recall, providing a balanced metric that accounts for both false positives and false negatives. It is especially valuable when dealing with imbalanced datasets, as it gives a single metric reflecting both the model's precision and recall.

In the above equations, TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives represent the classification outcomes.

The Compression Ratio (CR) tells us how much of the original data reduction during compression. It's calculated as:

$$\text{CR} = \left(1 - \frac{M}{N}\right) \times 100\% \tag{6.6}$$

where M is the size of the compressed data, and N is the original data size. In simple terms, a higher CR means more data has been compressed, while a lower CR indicates that most of the original data has been retained. In our study, we tested CR values from 10% to 90% to evaluate how different levels of compression affect classification accuracy. Higher compression (like 90%) makes the data smaller and faster to process but can impact accuracy, while lower compression like 10% keeps more data detail, which can help with accuracy but requires more resources. By experimenting with these different levels, we aimed to find a balance between efficient data use and reliable classification.

6.5 Results and Discussions

We analyzed the model performance of with different sensing matrix at various compression level then we compared the performance with traditional classifiers (SVM, k-NN and Neural Network).

6.5.1 Model Accuracy Across Sensing Matrices and Compression Ratios

In this chapter, we evaluated the accuracy of our model across four distinct sensing matrices—Random Gaussian Matrix (RGM), Random Binary Matrix (RBMB), Fourier Sensing Matrix (FSM), and Block Diagonal Matrix (BDM) at various compression ratios (CR) ranging from 10% to 90%. The performance results, as shown in Table 6.3, shows the meaningful differences in accuracy depending on both the choice of sensing matrix and the level of compression applied.

The Block Diagonal Matrix (BDM) consistently achieved the highest accuracy across all compression ratios, with performance remaining above 94% even at compression levels as high as 75%. This stability highlights the robustness of BDM in preserving data fidelity, even as data is increasingly compressed. At a 10% compression ratio, BDM reached an accuracy of 96.56%, and it maintained commendable accuracy of 89.99% even at the most extreme compression ratio of 90%.

The Fourier Sensing Matrix (FSM) also showed strong accuracy across most compression ratios, although it exhibited a slight drop as compression increased beyond 70%. Starting with an accuracy of 94.49% at 10% compression, FSM retained over 88% accuracy at 90% compression, which shows its reliability in preserving the features in compressed data although somewhat less consistently than BDM under high compression scenarios.

Random Binary Matrix (RBMB) and Random Gaussian Matrix (RGM) demonstrated lower accuracies compared to BDM and FSM, particularly as compression increased. For example, RBMB achieved an accuracy of 89.21% at 10% compression but declined to 78.30% at 90%. Similarly, RGM began with an accuracy of 86.67% at low compression but dropped to 76.70% at 90% compression. These results indicate that RBMB and RGM are more susceptible to accuracy degradation under higher compression levels, suggesting limitations in their ability to retain essential features as effectively as FSM and BDM due to their randomness behavior. After analyzing the accuracy at all sensing matrices, it shows that BTM as the optimal sensing matrix, consistently delivering the highest accuracy across compression ratios.

Table 6.3: Model accuracy (%) at different sensing matrices and various compression ratios

Sensing Matrix	CR-10%	CR-20%	CR-30%	CR-40%	CR-50%	CR-60%	CR-70%	CR-75%	CR-80%	CR-87.5%	CR-90%
	Accuracy %										
RGM	86.672	86.364	86.798	85.834	87.076	86.413	85.261	85.454	83.592	82.513	76.701
RBMB	89.210	87.896	89.794	87.908	87.841	88.155	87.263	86.039	85.087	80.909	78.299
FSM	94.490	94.762	94.985	94.882	94.810	92.254	91.452	91.428	91.778	90.482	88.348
BDM	96.564	96.775	95.569	94.768	96.540	95.967	95.286	95.298	95.129	93.273	89.999

6.5.2 Precision, Recall and F1-Score Analysis

In this section, we examine the model's performance in terms of precision, recall, and F1-score across various compression ratios and given sensing matrices. Figure 6.3 presents the results, where each metric is analyzed at compression ratios ranging from 10% to 90%. The precision results suggest that both BD and FSM consistently outperform the RGM and RBM across all compression ratios. At lower compression ratios (10% to 40%), BD achieves nearly perfect precision, maintaining above 96% even as compression increases. FSM follows closely, showing a slight decline as compression reaches higher levels but still maintaining competitive values compared to RBMB and RGM. In contrast, RGM and RBMB display lower precision, especially at higher compression ratios, where RGM's precision dips below 80% at 90% compression. This trend suggests that BDM and FSM are more effective in retaining significant features needed for precise predictions, even under hight compression.

Similarly, the recall results draw attention to the of BDM and FSM. Across all compression ratios, BDM sustains the highest recall values, exceeding 95% up to a compression ratio of 75%, before slightly dropping as compression becomes more aggressive. FSM also exhibits strong recall values, particularly at compression levels up to 60%. Conversely, RGM and RBM experience a more drop in recall as compression increases. By 90% compression, both matrices fall below 80% recall, indicating a loss of sensitivity to identifying relevant instances in the data at higher compression levels. Findings highlight the superiority of BDM and FSM in capturing the necessary patterns for recall at various compression stages.

For the F1-score, which provides a balanced measure of precision and recall, BDM and FSM again lead, demonstrating strength across most compression ratios. BDM maintains F1-scores above 95% up to 75% compression, showcasing its capacity to balance precision and recall effectively. FSM, while showing a minor decline in F1-score at higher compression levels, consistently outperforms RBMB and RGM, especially in the mid-compression range (40% to 75%). In contrast, RGM and RBM show a noticeable decline in F1-score as compression ratio increases, with RGM falling below 80% at 90% compression. This trend suggests that RGM and RBMB struggle to retain essential information as the data becomes more compressed, impacting the model's overall balance in performance.

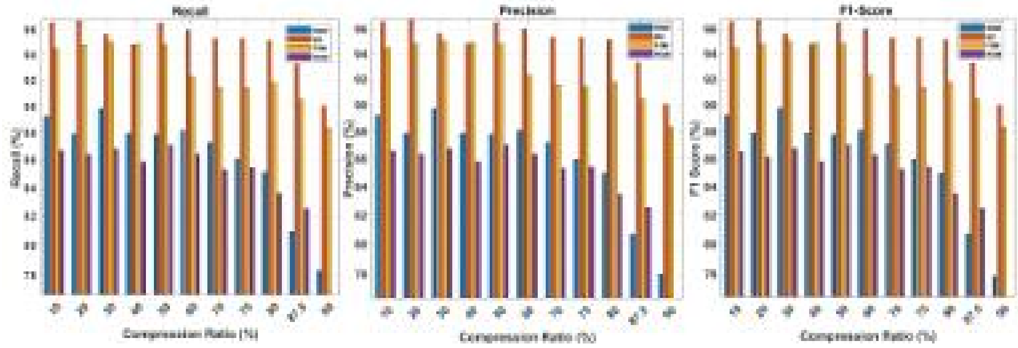

Figure 6.3: Presision, Reccall and Accuracy of The model across four sensing matrices at various compression level

6.5.3 Comparison with State-of-the-Art Classification Methods

From the above analysis, it is clear that the Block Diagonal (BD) sensing matrix consistently outperforms other matrices in terms of accuracy, particularly at higher compression ratios. Given this strong performance, we chose the BD matrix for compression and evaluated its compatibility with various state-of-the-art classifiers, including K-Nearest Neighbors (KNN), Neural Network (NN), Random Forest, Support Vector Machine (SVM), and XGBoost. Table 6.4 presents the accuracy of these classifiers across different compression ratios, comparing them directly with our proposed model.

As shown in Table 6.4, our proposed model consistently achieves the highest accuracy across all compression levels, outperforming traditional classifiers by a significant margin. At a low compression ratio of 10%, our model achieves an high accuracy of 96.56%, which surpasses even the highest-performing classifiers, such as KNN (90.34%) and XGBoost (89.34%). This trend continues as the compression ratio increases, with our model maintaining robust performance even at a 90% compression ratio, where it achieves an accuracy of 89.99%. In comparison, the next best performer, KNN, drops to 87.76%, while XGBoost falls further to 84.41%.

The neural network classifier shows moderate performance, with accuracy ranging from 83.34% at 10% compression to 79.17% at 90% compression. This suggests that while neural networks can handle compression reasonably well, they struggle to retain high accuracy at high compression levels. Similarly, Random Forest and SVM exhibit lower accuracy compared to our model, especially as the compression ratio increases. Random Forest peaks at around 77.24% accuracy at 90% compression, while SVM performs even lower, achieving only 74.80% accuracy at this level.

The results indicate that our proposed model combine with the BDM sensing matrix, demonstrates outstanding performance and adaptability to data compression. Unlike traditional classifiers, which tend to experience a noticeable decline in accuracy at higher compression levels, our model maintains high accuracy, suggesting that it effectively preserves essential data features even under significant compression.

Table 6.4: Accuracy Comparison with State-of-the Art Classifiers

Compression Ratio	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Models	**Accuracy %**								
KNN	90.343	90.253	89.704	89.017	90.054	90.295	90.403	90.017	87.763
Neural Network	83.338	81.470	80.873	78.878	84.194	83.067	83.133	81.952	79.167
Random Forest	75.791	75.586	75.845	75.405	75.598	75.327	75.231	75.725	77.238
SVM	80.180	80.156	79.703	78.673	79.679	79.529	78.474	76.810	74.797
XGBoost	89.336	89.023	88.715	87.920	87.588	86.817	86.298	85.955	84.405
Our Proposed	**96.564**	**96.775**	**95.569**	**94.768**	**96.540**	**95.967**	**95.286**	**95.129**	**89.999**

6.5.4 Hardware Implementation of the Optimal Sensing Matrix

The block diagonal binary matrix was selected for hardware implementation due to its efficient structure, which supports high data compression while retaining the critical features needed for accurate classification. It is important to reduce data dimensionality without sacrificing performance. The block diagonal matrix, with its binary structure and organized blocks of ones along the diagonal, meets these requirements effectively. Its structured design simplifies computations and leads to faster processing times, making it particularly suitable for real-time compression.

Figure 6.4 illustrates the layout of the block diagonal matrix, which consists of rows with sets of consecutive ones aligned diagonally. This regular block pattern not only simplifies computations for data compression but also minimizes memory requirements, as each row only activates a specific portion of the matrix. This design maintains consistent processing times across varying compression ratios, as shown in the experimental results.

To evaluate the performance on hardware, we implemented this matrix on the STM32 NUCLEO-F401RE microcontroller, which operates at a clock frequency of 84 MHz. The original signals were sampled at a frequency of 128 Hz, meaning each sample is captured every $\frac{1}{128} \approx 7.8125$ milliseconds. This time interval defines the available processing window between successive samples. For real-time systems, it is essential that the processing time falls well within this interval, ensuring that the system can keep up with incoming data without delays.

After measuring the processing time, we used the CubeMX power consumption calculator to estimate the power and energy usage during this period. Table 6.5 presents these results, including clock cycles, processing time in microseconds, and energy consumption in microjoules. Notably, the processing times range between 7 and 9 microseconds for various compression ratios, which is significantly less than the 7.8125 milliseconds sample interval. This leaves a substantial margin, allowing the microcontroller to enter low-power states between processing cycles, which can help conserve energy. For example, with an average processing time of 8 microseconds, the system has over 99.8% of each sampling interval available as idle time, which could contribute to reduced power usage. These results demonstrate that the block diagonal matrix can be implemented on hardware with minimal impact on processing speed and energy efficiency. Even at lower compression ratios, this structure ensures that processing times remain stable and manageable.

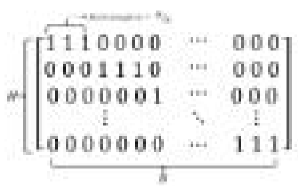

Figure 6.4: Block Diagonal Sensing Matrix

Table 6.5: Microcontroller Implementation of Block Diagonal Matrix

Microcontroller: STM32F401RE Nucleo with 84 MHz clock frequency					
Compression Ratio	**Clock Cycles**	**Processing Time (µs)**	**Current (mA)**	**Power (mW)**	**Energy (µJ)**
10	686	8	21.6	71.28	0.57024
20	650	8	21.6	71.28	0.57024
30	597	7	21.6	71.28	0.49896
40	571	7	21.6	71.28	0.49896
50	698	9	21.6	71.28	0.64152
60	627	8	21.6	71.28	0.57024
70	666	8	21.6	71.28	0.57024
75	684	8	21.6	71.28	0.57024
80	690	9	21.6	71.28	0.64152
87.5	683	8	21.6	71.28	0.57024
90	665	8	21.6	71.28	0.57024

6.6 Limitations and Future Work

While the proposed compressed learning framework for ECG classification demonstrates significant advantages in terms of computational efficiency and classification accuracy, although, there are few limitations that need to be addressed in future. First, we used only three different signals and databases in future which can be extended to various types of signals.

Another limitation involves the fixed structure of the block diagonal binary sensing matrix. Although it provides computational benefits and maintains high accuracy at various compression ratios, this matrix may not always optimally capture the most informative features from the signals, especially at high compression levels. Future work could involve developing adaptive or learned sensing matrices that adjust based on the signal characteristics, possibly to use neural network-based approaches to dynamically optimize the matrix structure during training.

The energy consumption and processing times observed during hardware implementation, while efficient, could benefit from further optimization. The STM32 NUCLEO-F401RE microcontroller used in this study offers a practical baseline, but more advanced, low-power microcontrollers and FPGA devices may achieve even greater power savings. Additionally, the developed classification model can be implemented on FPGA or Microcontroller for resource optimization .

6.7 Conclusion

This chapter presents a comprehensive approach to ECG classification using compressed learning, with a focus on optimizing computational efficiency and preserving classification accuracy. Through the use of a block diagonal binary matrix, we successfully compressed ECG signals while retaining the features necessary for accurate classification of arrhythmia (ARR), congestive heart failure (CHF), and normal sinus rhythm (NSR). Our framework take advantages of the structural simplicity of the block diagonal matrix, which enables efficient processing on hardware with limited resources. The implementation on an STM32 NUCLEO-F401RE microcontroller demonstrated the feasibility of the approach in real-time settings, with processing times well within the interval allowed by a 128 Hz sampling frequency, thus enabling timely data analysis without data backlog or delay.

The evaluation across multiple sensing matrices and compression ratios disclose that the block diagonal matrix offers superior classification accuracy, particularly at higher compression levels. The CNN-based classifier used in the framework consistently outperformed traditional classifiers like SVM, KNN, and XGBoost, maintaining accuracy even at compression ratios up to 90%. This performance demonstrates the effectiveness of compressed learning in achieving a balance between data reduction and diagnostic accuracy, meeting the urgent demand for efficient cardiac monitoring solutions in wearable and remote health systems.

Chapter 7

Conclusion and Future Recommendations

7.1 Conclusion

This thesis has explored innovative approaches to compressive sensing and compressed learning for ECG signal, addressing critical challenges such as data volume, energy efficiency, secure transmission, and computational complexity. By integrating theoretical advancements with practical implementations, the research provides a comprehensive framework that bridges the gap between abstract concepts and real-world healthcare applications.

Chapter 2 laid the theoretical groundwork, detailing the mathematical principles of CS and its relevance to sparse ECG signals. It critically reviewed existing CS methodologies for biomedical signals, highlighting gaps in reconstruction accuracy, energy efficiency, and scalability in healthcare applications. This review established a foundation for the novel frameworks proposed in subsequent chapters, focusing on tailored solutions for ECG signal acquisition, processing, and analysis.

In Chapter 3, the thesis introduced a dual-layer encryption system that integrated CS with column permutation to secure ECG data transmission. This framework takes advantages of the randomness inherent in CS for cryptographic purposes, ensuring robust data security without additional computational overhead. The hardware implementations demonstrated real-time compatibility and energy efficiency, with processing times as low as 1.03 milliseconds at CR=50% on FPGA platforms and power consumption well-suited for wearable devices. This contribution addressed the dual challenges of secure and efficient ECG signal processing in resource-constrained environments.

The data-driven sensing matrices were designed in chapter 4 to optimize feature extraction from ECG signals. Unlike traditional random matrices, the proposed matrix achieved superior compression and reconstruction performance, preserving diagnostic fidelity even at compression ratios (CRs) as high as 85%. The chapter also introduced a binary thresholding technique to transform continuous-valued sensing matrices into binary formats, enabling efficient implementation on hardware platforms. This innovation reduced computational complexity and energy consumption, ensuring scalability for real-world applications.

The integration of deep learning into CS frameworks was explored in Chapter 5, where end-to-end autoencoder-based models were proposed for simultaneous sensing and reconstruction. These models eliminated the need for iterative reconstruction, achieving high accuracy and computational efficiency. Hardware implementations of these deep learning models demonstrated their feasibility and making them a viable option for low-power wearable healthcare devices. The chapter established the potential of neural networks to enhance CS applications, setting a benchmark for future research.

In Chapter 6, the thesis shifted focus to compressed learning, introducing a novel paradigm for direct classification of ECG signals from compressed measurements. This approach bypassed the traditional reconstruction step, significantly reducing computational overhead while maintaining diagnostic reliability. Using a CNN-based classifier, the framework achieved high classification accuracy of 95% across various cardiac conditions, even at CRs of 75%. The use of block-diagonal binary sensing matrices ensured the preservation of essential features, enabling efficient and accurate classification directly in the compressed domain. This innovation represents a significant step forward in making real-time health monitoring systems more efficient and scalable.

Collectively, the contributions of this thesis address some of the most pressing challenges in ECG signal processing. The proposed frameworks not only improve compression and reconstruction but also introduce secure transmission methods and real-time classification techniques that are compatible with wearable and remote healthcare technologies. By integrating hardware implementations, the thesis ensures that these solutions are not merely theoretical but are ready for deployment in real-world scenarios.

Looking forward, the methodologies and insights developed in this work have the potential to transform the landscape of biomedical signal processing. From enabling long-term health monitoring with minimal energy consumption to advancing secure and efficient health data transmission, the contributions of this thesis pave the way for more accessible, reliable, and scalable healthcare solutions. These advancements underscore the promise of compressed sensing and learning as transformative tools in modern healthcare, addressing the growing demand for smarter, faster, and more energy-efficient diagnostic systems.

7.2 Future Recommendations

While this thesis introduces innovative solutions for ECG signal processing, it also opens several avenues for future research and development. These opportunities can further enhance the methodologies and extend their applicability to address broader challenges in biomedical signal processing. The following recommendations outline potential directions for this work:

One promising area for exploration lies in optimizing the proposed frameworks to handle higher compression ratios (CRs). While this thesis demonstrates robust performance at CRs up to 85%, achieving similar or better reconstruction quality at CRs exceeding 90% would represent a significant advancement. Future work could focus on developing advanced sparsity-driven algorithms or incorporating adaptive sensing matrices to ensure that the PRD remains below 5%, even at such extreme levels of data compression. These efforts would make the frameworks more efficient for scenarios requiring aggressive data reduction, such as real-time remote monitoring with limited bandwidth.

The techniques developed in this thesis, particularly the data-driven sensing matrix and compressed learning models, can be expanded to other physiological signals beyond ECG. Signals like electroencephalograms (EEG) and electromyograms (EMG) present unique challenges due to their distinct sparsity characteristics and noise profiles. Adapting the proposed methodologies to these signals would involve tailoring the sensing matrices and reconstruction algorithms to optimize performance, potentially enabling breakthroughs in brain-computer interfaces, neuroprosthetics, and muscle activity monitoring.

Another critical direction involves advancing feature detection capabilities within the compressed domain. While the thesis primarily focused on three signal classifications, future research could delve deeper into identifying complex patterns such as QRS complexes, P and T waves, or other abnormalities directly from compressed measurements. Developing algorithms that enable comprehensive feature extraction without requiring full signal reconstruction could enhance diagnostic accuracy while maintaining computational efficiency. This would also pave the way for real-time analysis systems capable of identifying a broader range of cardiac conditions.

The binary thresholding sensing matrix proposed in this work achieved excellent results with densities ranging from 1% to 15%. However, future investigations could explore higher density levels to assess their impact on reconstruction quality and computational efficiency. Higher densities may strike a better balance between preserving diagnostic fidelity and optimizing resource consumption, particularly for scenarios where slight increases in computational demands are acceptable.

Energy efficiency remains a critical concern in wearable and embedded systems, and this thesis demonstrated promising results in this domain. However, further improvements are achievable through advanced techniques such as quantization-aware training and lightweight deep learning architectures. Additionally, implementing the proposed framework on Field-Programmable Gate Arrays (FPGAs) could provide valuable insights into energy optimization, as FPGAs offer a unique combination of high performance and low power consumption. These steps would contribute significantly to extending the battery life of wearable devices, ensuring their reliability in long-term health monitoring applications.

Finally, the proposed compressed learning method holds significant potential for biometric identification and authentication. By leveraging the unique features of individual ECG signals, future research could refine these methods for creating robust, non-invasive biometric systems. Expanding this work to include multimodal biometric systems that integrate ECG with other physiological signals such as photoplethysmograms (PPG) data could enhance security and reliability. These systems would be particularly valuable in applications requiring secure and continuous authentication, such as remote healthcare platforms or personal health monitoring devices.

In summary, these recommendations highlight the potential for expanding and refining the methodologies presented in this thesis. By addressing higher compression ratios, exploring additional physiological signals, advancing feature detection, optimizing energy efficiency, and expanding to biometric applications, future work can build upon the strong foundation established here. These directions promise to further enhance the capabilities and impact of compressed sensing and learning in modern healthcare, paving the way for smarter, more efficient, and scalable biomedical technologies.

Publications During Ph.D.

1. F. Spagnolo, Bharat Lal, P. Corsonello, and R. Gravina, “A Novel Compressive Sensing Method for Secure and Energy Efficient ECG Signal Transmission Applications” in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2025.3530082, January 2025

2. Bharat Lal, Khushal Das, Miguel Heredia Conde, Pasquale Corsonello, Raffaele Gravina, “Comparative Analysis of Compressive Sensing Reconstruction Algorithms for ECG Signals” in 6th International Workshop on the Theory of Computational Sensing and its applications to Radar, Multimodal Sensing, and Imaging (CoSeRa 2024).

3. Qimeng Li, Bharat Lal, and Raffaele Gravina, “Multi-user Activity Monitoring based on Contactless Sensing” in Activity Recognition and Prediction for Smart IoT Environments. Cham: Springer Nature Switzerland, 2024. 97-110.

4. Bharat Lal, R. Gravina, F. Spagnolo, and P. Corsonello, ”Compressed Sensing Approach for Physiological Signals: A Review” in IEEE Sensors Journal, doi: 10.1109/JSEN.2023.3243390.

5. Bharat Lal, Qimeng Li, Raffaele Gravina, and Pasquale Corsonello, “Compressed Sensing-based IoMT Applications” in Device-Edge-Cloud Continuum Internet of Things, Springer Nature Switzerland, November 2023.

6. Bharat Lal, Q. Li, R. Gravina, and P. Corsonello, “Abnormal ECG Detection in Wearable Devices Using Compressed Learning” in 20th IEEE International Conference on Networking, Sensing and Control, 25-27 October 2023, Marseille, France.

7. Bharat Lal, Miguel Heredia Conde, Pasquale Corsonello, and Raffaele Gravina, “Secure and Energy-Efficient ECG Signal Monitoring in the IoT Healthcare using Compressive Sensing” in 21st IEEE International Conference on Dependable, Autonomic & Secure Computing, 24-27 November 2023, Dubai, UAE.

8. Bharat Lal, Q. Li, R. Gravina, and P. Corsonello, ”ECG Signals Analysis based on Compressed Sensing and Learning Techniques for Heart Disease Recognition” in 2023 7th International Multi-Topic ICT Conference (IMTIC), Jamshoro, Pakistan, 2023, pp. 1-7, doi: 10.1109/IMTIC58887.2023.10178468.

9. Bharat Lal, (2023, May 30). Compressed Sensing and Learning for Secure and Low-Power Health Care Devices. Poster presented at Ph.D. Day, University of Calabria, Italy.

10. Bharat Lal, (2022, September 12). Compressed Sensing based Physiological Signals Acquisitions and Features Extraction. Poster presented at the 1st International School on Internet of Things & Edge AI: Computing, Communications, and Systems.

Bibliography

[1] Andrew Callaway, “The healthcare data explosion.” RBC Capital Markets, Global Investment Banking, 2023. Accessed: 2023-07-28.

[2] H. Nyquist, “Certain topics in telegraph transmission theory,” *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[3] C. E. Shannon, “Communication in the presence of noise,” *Proceeding of the Institute of Radio Engineers*, 1949.

[4] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[5] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[6] E. J. Candes, J. Romberg, and T. Tao, “Compressive sampling,” *Proceedings of the international congress of mathematicians*, vol. 3, pp. 1433–1452, 2006.

[7] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[8] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[9] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] A. A. Picorone, T. R. de Oliveira, R. Sampaio-Neto, M. Khosravy, and M. V. Ribeiro, “Channel characterization of low voltage electric power distribution

networks for plc applications based on measurement campaign," *International Journal of Electrical Power & Energy Systems*, vol. 116, p. 105554, 2020.

[11] S.-x. Nan, X.-f. Feng, Y.-f. Wu, and H. Zhang, "Remote sensing image compression and encryption based on block compressive sensing and 2d-lcccm," *Nonlinear dynamics*, vol. 108, no. 3, pp. 2705–2729, 2022.

[12] Y. Sun, J. Chen, Q. Liu, B. Liu, and G. Guo, "Dual-path attention network for compressed sensing image reconstruction," *IEEE Transactions on Image Processing*, vol. 29, pp. 9482–9495, 2020.

[13] V. Kilic, T. D. Tran, and M. A. Foster, "Compressed sensing in photonics: tutorial," *Journal of the Optical Society of America B*, vol. 40, no. 1, pp. 28–52, 2022.

[14] D. Gurve, D. Delisle-Rodriguez, T. Bastos-Filho, and S. Krishnan, "Trends in compressive sensing for EEG signal processing applications," *Sensors*, vol. 20, no. 13, p. 3703, 2020.

[15] J. Bobin, J.-L. Starck, and R. Ottensamer, "Compressed sensing in astronomy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 718–726, 2008.

[16] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *CVPR 2011*, pp. 457–464, IEEE, 2011.

[17] 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.

[18] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[19] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "On the interplay between routing and signal representation for compressive sensing in wireless sensor networks," in *2009 Information Theory and Applications Workshop*, pp. 206–215, IEEE, 2009.

[20] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[21] Z. Tian and G. B. Giannakis, "Compressed sensing for wideband cognitive radios," in *2007 IEEE International conference on acoustics, speech and signal processing-ICASSP'07*, vol. 4, pp. IV–1357, IEEE, 2007.

[22] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 798–805, IEEE, 2008.

[23] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[24] K. Lee and Y. Bresler, "Admira: Atomic decomposition for minimum rank approximation," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4402–4416, 2010.

[25] A. B. M. Aharon, M. Elad, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[26] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: asymptopia?," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 2, pp. 301–337, 1995.

[27] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge university press, 2012.

[28] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[29] Q. Yan, Y. Xu, X. Yang, and T. Q. Nguyen, "Single image superresolution based on gradient profile sharpness," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3187–3202, 2015.

[30] I. Tawfic and S. Kayhan, "Compressed sensing of ECG signal for wireless system with new fast iterative method," *Computer methods and programs in biomedicine*, vol. 122, no. 3, pp. 437–449, 2015.

[31] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[32] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[33] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[34] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[35] J. A. Tropp, "Recovery of short, complex linear combinations via l1 minimization," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1568–1570, 2005.

[36] J. A. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[37] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[38] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[39] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[40] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond nyquist: Efficient sampling of sparse bandlimited signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, 2010.

[41] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Communications of the ACM*, vol. 53, no. 12, pp. 93–100, 2010.

[42] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[43] Z. Bai, T. Wimalajeewa, Z. Berger, G. Wang, M. Glauser, and P. K. Varshney, "Low-dimensional approach for reconstruction of airfoil data via compressive sensing," *AIAA Journal*, vol. 53, no. 4, pp. 920–933, 2014.

[44] I. Bright, G. Lin, and J. N. Kutz, "Compressive sensing and machine learning strategies for characterizing the flow around a cylinder with limited pressure measurements," *Physics of Fluids*, vol. 25, no. 127102, pp. 1–15, 2013.

[45] I. Bright, G. Lin, and J. N. Kutz, "Classification of spatio-temporal data via asynchronous sparse sampling: Application to flow around a cylinder," *arXiv preprint arXiv:1506.00661*, 2015.

[46] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, "Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems," *SIAM Journal on Applied Dynamical Systems*, vol. 13, no. 4, pp. 1716–1732, 2014.

[47] J. N. Kutz, *Data-Driven Modeling Scientific Computation: Methods for Complex Systems Big Data*. Oxford University Press, 2013.

[48] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, pp. 189–206, 1984.

[49] J. E. Fowler, "Compressive-projection principal component analysis," *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2230–2242, 2009.

[50] A. C. Gilbert, J. Y. Park, and M. B. Wakin, "Sketched SVD: Recovering spectral features from compressive measurements," *ArXiv e-prints*, 2012.

[51] H. Qi and S. M. Hughes, "Invariance of principal components under low-dimensional random projection of the data," in *IEEE International Conference on Image Processing*, pp. 1–4, 2012.

[52] World Health Organization, "Cardiovascular diseases (CVDs)," 2024. Accessed: 2024-11-13.

[53] D. Mitra, H. Zanddizari, and S. Rajan, "Investigation of kronecker-based recovery of compressed ECG signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3642–3653, 2019.

[54] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, "Energy-efficient ECG compression on wireless biosensors via minimal coherence sensing and weighted l1 minimization reconstruction," *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 520–528, 2014.

[55] F. M. Dias, M. Khosravy, T. W. Cabral, H. L. M. Monteiro, L. M. de Andrade Filho, L. de Mello Honório, R. Naji, and C. A. Duque, "Compressive sensing of electrocardiogram," in *Compressive sensing in healthcare*, pp. 165–184, Elsevier, 2020.

[56] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion WDD measure for ECG signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, 2000.

[57] G. Da Poian, R. Bernardini, and R. Rinaldo, "Separation and analysis of fetal-ECG signals from compressed sensed abdominal ECG recordings," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1269–1279, 2015.

[58] T. Y. Rezaii, S. Beheshti, M. Shamsi, and S. Eftekharifar, "ECG signal compression and denoising via optimum sparsity order selection in compressed sensing framework," *Biomedical Signal Processing and Control*, vol. 41, pp. 161–171, 2018.

[59] M. Rakshit and S. Das, "Electrocardiogram beat type dictionary based compressed sensing for telecardiology application," *Biomedical Signal Processing and Control*, vol. 47, pp. 207–218, 2019.

[60] D. Bortolotti, M. Mangia, A. Bartolini, R. Rovatti, G. Setti, and L. Benini, "Energy-aware bio-signal compressed sensing reconstruction on the WBSN-gateway," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 3, pp. 370–381, 2016.

[61] G. Da Poian, C. J. Rozell, R. Bernardini, R. Rinaldo, and G. D. Clifford, "Matched filtering for heart rate estimation on compressive sensing ECG measurements," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 6, pp. 1349–1358, 2017.

[62] D. Yoon, H. S. Lim, K. Jung, T. Y. Kim, and S. Lee, "Deep learning-based electrocardiogram signal noise detection and screening model," *Healthcare informatics research*, vol. 25, no. 3, pp. 201–211, 2019.

[63] Cables and Sensors, "12-lead ECG placement guide with illustrations." https://www.cablesandsensors.eu/pages/12-lead-ECG-placement-guide-with-illustrations, 2024. Accessed: 2024-08-04.

[64] 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.

[65] E. Farnell, H. Kvinge, J. P. Dixon, J. R. Dupuis, M. Kirby, C. Peterson, E. C. Schundler, and C. W. Smith, "A data-driven approach to sampling matrix selection for compressive sensing," in *Computational Imaging V*, vol. 11396, p. 1139603, SPIE, 2020.

[66] D. Ramalho, K. Melo, M. Khosravy, F. Asharif, M. S. S. Danish, and C. A. Duque, "A review of deterministic sensing matrices," *Compressive Sensing in Healthcare*, pp. 89–110, 2020.

[67] Y. Yang, H. Liu, and J. Hou, "A compressed sensing measurement matrix construction method based on TDMA for wireless sensor networks," *Entropy*, vol. 24, no. 4, p. 493, 2022.

[68] H. Rauhut, "Compressive sensing and structured random matrices," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, no. 1, p. 92, 2010.

[69] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A systematic review of compressive sensing: Concepts, implementations and applications," *IEEE access*, vol. 6, pp. 4875–4894, 2018.

[70] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2022.

[71] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[72] L. Zhao, Y. Hu, and Y. Liu, "Stochastic gradient matching pursuit algorithm based on sparse estimation," *Electronics*, vol. 8, no. 2, p. 165, 2019.

[73] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on signal processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[74] A. L. Pilastri and J. M. R. Tavares, "Reconstruction algorithms in compressive sensing: An overview," in *11th edition of the Doctoral Symposium in Informatics Engineering (DSIE-16)*, 2016.

[75] R. Manchanda and K. Sharma, "A review of reconstruction algorithms in compressive sensing," in *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, pp. 322–325, IEEE, 2020.

[76] S. S. Kumar and P. Ramachandran, "Review on compressive sensing algorithms for ECG signal for IoT based deep learning framework," *Applied Sciences*, vol. 12, no. 16, p. 8368, 2022.

[77] A. Mishra, F. Thakkar, C. Modi, and R. Kher, "ECG signal compression using compressive sensing and wavelet transform," in *Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (San Diego, CA, USA), pp. 3404–3407, 2012.

[78] Y. C. Eldar, P. Sidorenko, D. G. Mixon, S. Barel, and O. Cohen, "Sparse phase retrieval from short-time fourier measurements," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 638–642, 2014.

[79] F. Bergeaud and S. Mallat, "Matching pursuit: Adaptive representations of images and sounds," *Computational and Applied Mathematics*, vol. 15, pp. 97–110, 1996.

[80] J.-L. Starck, E. Candes, and D. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

[81] D. L. Donoho and X. Huo, "Beamlet pyramids: A new form of multiresolution analysis suited for extracting lines, curves, and objects from very noisy image data," in *Proceedings of the SPIE International Symposium on Optical Science and Technology*, (San Diego, CA, USA), pp. 434–444, 2000.

[82] J.-L. Starck, D. L. Donoho, and E. J. Candès, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, pp. 670–684, 2002.

[83] 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.

[84] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," *preprint*, 2009.

[85] H.-T. Li, C.-Y. Chou, Y.-T. Chen, S.-H. Wang, and A.-Y. Wu, "Robust and lightweight ensemble extreme learning machine engine based on eigenspace domain for compressed learning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 12, pp. 4699–4712, 2019.

[86] J. Y. Park, M. B. Wakin, and A. C. Gilbert, "Modal analysis with compressive measurements," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1655–1670, 2014.

[87] M. Rani, S. B. Dhok, and R. B. Deshmukh, "EEG seizure detection from compressive measurements," in *Advances in VLSI, Communication, and Signal Processing* (D. Dutta, H. Kar, C. Kumar, and V. Bhadauria, eds.), (Singapore), pp. 963–969, Springer Singapore, 2020.

[88] S. Qiu, Z. Li, W. He, L. Zhang, C. Yang, and C.-Y. Su, "Brain–machine interface and visual compressive sensing-based teleoperation control of an exoskeleton robot," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 58–69, 2017.

[89] C.-Y. Chou, Y.-W. Pua, T.-W. Sun, and A.-Y. A. Wu, "Compressed-domain ECG-based biometric user identification using compressive analysis," *Sensors*, vol. 20, no. 11, 2020.

[90] Y. Wencheng and W. Song, "A privacy-preserving ECG-based authentication system for securing wireless body sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6148–6158, 2022.

[91] A. Suresh Kumar, M. Amin Salih, N. Kalpana, R. Kanagachidambaresan, G. S. B., V. Chaman, M. Traian Candin, and S. Calin Ovidiu, "A novel energy efficient threshold based algorithm for wireless body sensor network," *Energies*, vol. 15, no. 6095, 2022.

[92] A. Vidyadhar Jinnappa, D. Vijaypal Singh, P. Anubha, K. Sunil, and R. Imad, "Internet of things in healthcare: A survey on protocol standards, enabling technologies, wban architectures and open issues," *Physical Communication*, vol. 60, no. 102103, 2023.

[93] B. Lal, Q. Li, R. Gravina, and P. Corsonello, "Compressed sensing-based IoMT applications," in *Device-Edge-Cloud Continuum: Paradigms, Architectures and Applications*, pp. 183–202, Springer, 2023.

[94] B. Lal, R. Gravina, F. Spagnolo, and P. Corsonello, "Compressed sensing approach for physiological signals: A review," *IEEE Sensors Journal*, pp. 1–1, 2023.

[95] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Compressed sensing for bioelectric signals: A review," *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 529–540, 2014.

[96] M. Hossein, K. Nadia, A. David, and V. Pierre, "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.

[97] Z. Yushu, X. Yong, Z. Leo Yu, R. Yue, and G. Song, "Secure wireless communications based on compressive sensing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1093–1111, 2023.

[98] B. Lal, M. H. Conde, P. Corsonello, and R. Gravina, "Secure and energy-efficient ECG signal monitoring in the IoT healthcare using compressive sensing," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 0333–0339, 2023.

[99] M. Fira and L. Goras, "On projection matrices and dictionaries in ECG compressive sensing - a comparative study," in *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, pp. 3–8, 2014.

[100] A. Mishra, F. N. Thakkar, C. Modi, and R. Kher, "Selecting the most favorable wavelet for compressing ECG signals using compressive sensing approach," in *2012 International Conference on Communication Systems and Network Technologies*, pp. 128–132, IEEE, 2012.

[101] L. N. Sharma, “Multiscale entropy based compressive sensing for electrocardiogram signal compression,” in *2013 Eleventh International Conference on ICT and Knowledge Engineering*, pp. 1–5, 2013.

[102] A. Singh, J. J. Nallikuzhy, and S. Dandapat, “Compressed sensing framework of data reduction at multiscale level for eigenspace multichannel ECG signals,” in *2015 Twenty First National Conference on Communications (NCC)*, pp. 1–6, 2015.

[103] M. I. Chidean, Ó. Barquero-Pérez, Q. Zhang, R. H. Jacobsen, and A. J. Caamaño, “High diagnostic quality ECG compression and cs signal reconstruction in body sensor networks,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6255–6259, IEEE, 2016.

[104] F. Picariello, G. Iadarola, E. Balestrieri, I. Tudosa, and L. D. Vito, “A novel compressive sampling method for ECG wearable measurement systems,” *Measurement*, vol. 167, no. 108259, 2021.

[105] Y. Yang, F. Huang, F. Long, and Y. Tang, “Design of an adaptive ECG signal processing system based on compressed sensing,” in *5th International Conference on Universal Village (UV)*, 2020.

[106] 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.

[107] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, “Compressed sensing based method for ECG compression,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 761–764, IEEE, 2011.

[108] O. Kováč, J. Kromka, J. Šaliga, and A. Jusková, “Multiwavelet-based ECG compressed sensing,” *Measurement*, vol. 220, p. 113393, 2023.

[109] S. Abhishek, S. Veni, and K. Narayanankutty, “Biorthogonal wavelet filters for compressed sensing ECG reconstruction,” *Biomed. Signal Process. Control*, vol. 47, pp. 183–195, 2019.

[110] V. Izadi, P. K. Shahri, and H. Ahani, “A compressed-sensing-based compressor for ECG,” *Biomedical engineering letters*, vol. 10, pp. 299–307, 2020.

[111] A. M. A. Hassan, S. Mohsen, and M. M. Abo-Zahhad, “ECG signals compression using dynamic compressive sensing technique toward IoT applications,” *Multimedia Tools and Applications*, pp. 1–18, 2023.

[112] J. Šaliga, I. Andráš, P. Dolinskỳ, L. Michaeli, O. Kováč, and J. Kromka, "ECG compressed sensing method with high compression ratio and dynamic model reconstruction," *Measurement*, vol. 183, p. 109803, 2021.

[113] Y. V. Parkale and S. Nalbalwar, "Investigation on daubechies wavelet-based compressed sensing matrices for ECG compression," in *Computing, Communication and Signal Processing: Proceedings of ICCASP 2018*, pp. 707–716, Springer, 2019.

[114] R. R. Tamboli, M. A. Savkoor, S. Jana, and R. Manthalkar, "On the sparsest representation of electrocardiograms," in *Computing in Cardiology 2013*, pp. 479–482, 2013.

[115] Z. Li, Y. Deng, H. Huang, and S. Misra, "ECG signal compressed sensing using the wavelet tree model," in *2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)*, pp. 194–199, 2015.

[116] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden markov tree model," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5137–5140, IEEE, 2008.

[117] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488–3497, 2009.

[118] G. Plonka and J. Ma, "Curvelet-wavelet regularized split bregman iteration for compressed sensing," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 9, no. 01, pp. 79–110, 2011.

[119] A. M. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and emg wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[120] G. D. Poian, D. Brandalise, R. Bernardini, and R. Rinaldo, "Energy and quality evaluation for compressive sensing of fetal electrocardiogram signals," *Sensors 2017*, vol. 17, no. 9, pp. 1–13.

[121] H. Djelouat, X. Zhai, M. A. Disi, A. Amira, and F. Bensaali, "System-on-chip solution for patients biometric: a compressive sensing-based approach," *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9629–9639, 2018.

[122] H. Djelouat, A. Amira, F. Bensaali, and I. Boukhennoufa, "Secure compressive sensing for ECG monitoring," *Computers & Security*, vol. 88, no. 23, 2020.

[123] L. Li, L. Liu, H. Peng, Y. Yang, and S. Cheng, "Flexible and secure data transmission system based on semitensor compressive sensing in wireless body area networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3212–3227, 2018.

[124] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[125] 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.

[126] J.-H. Hsieh, K.-C. Hung, J.-H. Liu, and T.-C. Wu, "Wavelet-based quality-constrained ECG data compression system without decoding process," *IEEE MultiMedia*, vol. 27, no. 2, 2020.

[127] S. Banerjee and G. Singh, "Quality guaranteed ECG signal compression using tunable-q wavelet transform and möbius transform-based AFD," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 4008211–4008211, 2021.

[128] J. Shi, F. Wang, M. Qin, A. Chen, W. Liu, J. He, H. Wang, S. Chang, and Q. Huang, "New ECG compression method for portable ECG monitoring system merged with binary convolutional auto-encoder and residual error compensation," *Biosensors*, vol. 12, no. 524, 2022.

[129] G. Kulpeed and Q. Zhang, "Lightweight electrocardiogram signal compression," *Biomedical Signal Processing and Control*, vol. 85, no. 105012, 2023.

[130] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (WDD) measure for ECG signal compression," *IEEE transactions on biomedical engineering*, vol. 47, no. 11, pp. 1422–1430, 2000.

[131] M. Manikandan and S. Dandapat, "Wavelet energy based diagnostic distortion measure for ECG," *Biomedical Signal Processing and Control*, vol. 2, pp. 80–96, 2007.

[132] M. Fira, "Applications of compressed sensing: Compression and encryption," in *2015 E-Health and Bioengineering Conference (EHB)*, pp. 1–4, 2015.

[133] H. Zanddizari, S. Rajan, H. Zarrabi, and H. Rabah, "Privacy assured recovery of compressively sensed ECG signals," *IEEE Access*, vol. 10, pp. 17122–17133, 2022.

[134] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "On known-plaintext attacks to a compressed sensing-based encryption: A quantitative analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2182–2195, 2015.

[135] G. Hu, D. Xiao, Y. Wang, and T. Xiang, "An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 116–127, 2017.

[136] B. Lal, R. Gravina, F. Spagnolo, and P. Corsonello, "Compressed sensing approach for physiological signals: A review," *IEEE Sensors Journal*, 2023.

[137] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Adaptive dictionary reconstruction for compressed sensing of ECG signals," *IEEE journal of biomedical and health informatics*, vol. 21, no. 3, pp. 645–654, 2016.

[138] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, "Exploiting prior knowledge in compressed sensing wireless ECG systems," *IEEE journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 508–519, 2014.

[139] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza, "A real-time compressed sensing-based personal electrocardiogram monitoring system," in *2011 Design, Automation & Test in Europe*, pp. 1–6, IEEE, 2011.

[140] Z. Zhang, X. Liu, S. Wei, H. Gan, F. Liu, Y. Li, C. Liu, and F. Liu, "Electrocardiogram reconstruction based on compressed sensing," *IEEE Access*, vol. 7, pp. 37228–37237, 2019.

[141] Y. V. Parkale and S. L. Nalbalwar, "Application of compressed sensing (CS) for ECG signal compression: A review," in *Proceedings of the International Conference on Data Engineering and Communication Technology: ICDECT 2016, Volume 2*, pp. 53–65, Springer, 2017.

[142] M. Fira, L. Goras, C. Barabasa, and N. Cleju, "On ECG compressed sensing using specific overcomplete dictionaries," *Advances in Electrical and Computer Engineering*, vol. 10, no. 4, pp. 23–28, 2010.

[143] M. Fira, V. Maiorescu, and L. Goras, "The analysis of the specific dictionaries for compressive sensing of EEG signals," in *Proceedings of the Ninth International Conference on Advances in Computer-Human Interactions, Venice, Italy*, pp. 24–28, 2016.

[144] J. Chen, S. Sun, N. Bao, Z. Zhu, and L.-B. Zhang, "Improved reconstruction for CS-based ECG acquisition in internet of medical things," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25222–25233, 2021.

[145] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Compressed sensing: A simple deterministic measurement matrix and a fast recovery algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, 2015.

[146] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE transactions on biomedical engineering*, vol. 50, no. 3, pp. 289–294, 2003.

[147] G. D. Clifford, "A novel framework for signal representation and source separation: Applications to filtering and segmentation of biosignals," *Journal of Biological Systems*, vol. 14, no. 02, pp. 169–183, 2006.

[148] M. Fira, H.-N. Costin, and L. Goraș, "A study on dictionary selection in compressive sensing for ECG signals compression and classification," *Biosensors*, vol. 12, no. 3, p. 146, 2022.

[149] G. Da Poian, R. Bernardini, and R. Rinaldo, "Gaussian dictionary for compressive sensing of the ECG signal," in *2014 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings*, pp. 80–85, 2014.

[150] "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, author=Aharon, Michal and Elad, Michael and Bruckstein, Alfred, journal=IEEE Transactions on signal processing, volume=54, number=11, pages=4311–4322, year=2006, publisher=IEEE,"

[151] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale regularized least squares," *IEEE journal of selected topics in signal processing*, vol. 1, no. 4, pp. 606–617, 2007.

[152] J. Hua, H. Zhang, J. Liu, Y. Xu, and F. Guo, "Direct arrhythmia classification from compressive ECG signals in wearable health monitoring system," *J. Circuit. Syst. Comp.*, vol. 27, p. 1850088, 2018.

[153] J. Hua, J. Tang, J. Liu, F. Yang, and W. Zhu, "A novel ECG heartbeat classification approach based on compressive sensing in wearable health monitoring system," in *Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, (Atlanta, GA, USA), pp. 581–586, 2019.

[154] U. Gamper, P. Boesiger, and S. Kozerke, “Compressed sensing in dynamic mri,” *Magn. Reson. Med.*, vol. 59, pp. 365–373, 2010.

[155] Y. C. Eldar, P. Sidorenko, D. G. Mixon, S. Barel, and O. Cohen, “Sparse phase retrieval from short-time fourier measurements,” *IEEE Signal Process. Lett.*, vol. 22, pp. 638–642, 2015.

[156] M. N. Do and M. Vetterli, “Contourlets: A new directional multiresolution image representation,” in *Proceedings of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA, USA), pp. 497–501, 2002.

[157] M. Aharon, M. Elad, and A. M. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.*, vol. 54, pp. 4311–4322, 2006.

[158] A. G. Dimakis, R. Smarandache, and P. O. Vontobel, “LDPC codes for compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 58, pp. 3093–3114, 2010.

[159] L. Yu, J.-P. Barbot, G. Zheng, and H. Sun, “Compressive sensing with chaotic sequence,” *IEEE Signal Process. Lett.*, vol. 17, pp. 731–734, 2010.

[160] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, “Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse bayesian learning,” *IEEE Trans. Biomed. Eng.*, vol. 60, pp. 300–309, 2013.

[161] V. Izadi, P. K. Shahri, and H. Ahani, “A compressed-sensing-based compressor for ECG,” *Biomed. Eng. Lett.*, vol. 10, pp. 299–307, 2020.

[162] Z. Zhang, X. Liu, S. Wei, *et al.*, “Electrocardiogram reconstruction based on compressed sensing,” *IEEE Access*, vol. 7, pp. 37228–37237, 2019.

[163] M. Melek and A. Khattab, “ECG compression using wavelet-based compressed sensing with prior support information,” *Biomed. Signal Process. Control*, vol. 68, p. 102786, 2021.

[164] J. A. Jahanshahi, H. Danyali, and M. S. Helfroush, “Compressive sensing based multi-channel ECG reconstruction in wireless body sensor networks,” *Biomed. Signal Process. Control*, vol. 61, p. 102047, 2020.

[165] J. Saliga, I. Andráš, P. Dolinský, *et al.*, “ECG compressed sensing method with high compression ratio and dynamic model reconstruction,” *Measurement*, vol. 183, p. 109803, 2021.

[166] T. Y. Rezaii, S. Beheshti, M. Shamsi, *et al.*, "ECG signal compression and denoising via optimum sparsity order selection in compressed sensing framework," *Biomed. Signal Process. Control*, vol. 41, pp. 161–171, 2018.

[167] C. Yang, P. Pan, and Q. Ding, "Image encryption scheme based on mixed chaotic bernoulli measurement matrix block compressive sensing," *Entropy*, vol. 24, p. 273, 2022.

[168] L. F. Polania and R. I. Plaza, "Compressed sensing ECG using restricted boltzmann machines," *Biomed. Signal Process. Control*, vol. 45, pp. 237–245, 2018.

[169] P. R. Muduli, R. R. Gunukula, and A. Mukherjee, "A deep learning approach to fetal-ECG signal reconstruction," in *Proceedings of the Twenty Second National Conference on Communication (NCC)*, (Guwahati, India), pp. 1–6, 2016.

[170] M. Mangia, L. Prono, A. Marchioni, *et al.*, "Deep neural oracles for short-window optimized compressed sensing of biosignals," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, pp. 545–557, 2020.

[171] R. R. Shrivastwa, V. Pudi, C. Duo, *et al.*, "A brain–computer interface framework based on compressive sensing and deep learning," *IEEE Consum. Electr. Mag.*, vol. 9, pp. 90–96, 2020.

[172] M. Mangia, A. Marchioni, L. Prono, *et al.*, "Low-power ECG acquisition by compressed sensing with deep neural oracles," in *Proceedings of the 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, (Genova, Italy), pp. 158–162, 2020.

[173] R. R. Shrivastwa, V. Pudi, and A. Chattopadhyay, "An FPGA-based brain computer interfacing using compressive sensing and machine learning," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, (Hong Kong, China), pp. 726–731, 2018.

[174] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[175] Z. Zhang and B. D. Rao, "Extension of sbl algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009–2015, 2013.

[176] STMicroelectronics, "St edge ai - ai development tools for embedded systems," 2024. Accessed: 2024-11-11.

[177] M. Abdelazez, S. Rajan, and A. D. C. Chan, "Transfer learning for detection of atrial fibrillation in deterministic compressive sensed ECG," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 5398–5401, 2020.

[178] M. Abdelazez, F. F. Firouzeh, S. Rajan, and A. D. C. Chan, "Multi-stage detection of atrial fibrillation in compressively sensed electrocardiogram," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, 2020.

[179] G. Da Poian, C. Liu, R. Bernardini, R. Rinaldo, and G. D. Clifford, "Atrial fibrillation detection on compressed sensed ECG," *Physiological measurement*, vol. 38, no. 7, p. 1405, 2017.

[180] H. Zhang, Z. Dong, M. Sun, H. Gu, and Z. Wang, "TP-CNN: A detection method for atrial fibrillation based on transposed projection signals with compressed sensed ECG," *Computer Methods and Programs in Biomedicine*, vol. 210, p. 106358, 2021.

[181] A. Siddique, O. Hasan, F. Khalid, and M. Shafique, "Approxcs: near-sensor approximate compressed sensing for IoT-healthcare systems," *arXiv preprint arXiv:1811.07330*, 2018.

[182] G. Laudato, R. Oliveto, S. Scalabrino, A. R. Colavita, L. De Vito, F. Picariello, and I. Tudosa, "Identification of R-peak occurrences in compressed ECG signals," in *2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–6, 2020.

[183] H. Zhang, Z. Dong, J. Gao, P. Lu, and Z. Wang, "Automatic screening method for atrial fibrillation based on lossy compression of the electrocardiogram signal," *Physiological Measurement*, vol. 41, no. 7, p. 075005, 2020.

[184] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "A performance comparison of measurement matrices in compressive sensing," *International Journal of Communication Systems*, vol. 31, no. 10, p. e3576, 2018.

[185] D. S. Baim, W. S. Colucci, E. S. Monrad, H. S. Smith, R. F. Wright, A. Lanoue, D. F. Gauthier, B. J. Ransil, W. Grossman, and E. Braunwald, "Survival of patients with severe congestive heart failure treated with oral milrinone," *Journal of the American College of Cardiology*, vol. 7, no. 3, pp. 661–670, 1986.