PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Yahya Fares Medea
Faculty of Science
Department of Mathematics and Computer Science



Final Project

Submitted for obtaining Master Degree in Computer Science

Option: Systems Engineering And Web Technologies

Entitled :

Deep-ECG : Deep learning framework for ECG signal classification

Proposed and directed by:
Dr. KHELDOUN Ahmed

Prepared and presented by:
ABED Soufiane
REBAI Abdeldjalil

Academic Year : 2025-2026

ACKNOWLEDGEMENTS

First, we thank God Almighty, who granted us the strength and patience to complete this work and made this knowledge beneficial.
Second, we extend our sincere thanks to our supervisor, Professor **Dr. KHELDOUN Ahmed**, for his invaluable guidance, constant support, and assistance throughout the project.
We are also grateful to all the professors who guided us during our studies and shared their valuable expertise. Finally, we express our deep gratitude to everyone who contributed to this project, whether directly or indirectly.

Dedication

I dedicate this work to my dear family, and most especially to my beloved parents, for their unwavering support throughout my educational journey, their generosity, their tireless efforts in my studies, and their constant encouragement.

To my brothers and sisters: my eldest brother Halim, Amine, Hanane, and Nadjiya.

To my nephews and nieces: Ines, Imad, Nassim, Abd elJalile, Arwa, Kawthar, Tasnim, Israa, Ramzi, and my youngest niece, Nada.

To my colleague, **Abdeldjalil**.

To all the relatives I have mentioned and to those I may have forgotten, please accept my sincere apologies.

Soufiane

Dedication

I dedicate this modest work to my dear family, and most particularly to my beloved parents. Thank you for your unconditional support throughout my academic journey, for your kindness, your constant efforts, and your invaluable encouragement.

To my dear partner, **Soufiane**, for this collaboration and teamwork.

To all my friends, for their companionship, shared moments, and continuous support.

Abdeldjalil

Abstract

Electrocardiography (ECG) remains the standard method for identifying cardiac irregularities. Automatic detection of abnormalities from the ECG can aid in the early detection, diagnosis, and prevention of cardiovascular diseases. This work presents a novel deep-learning system, named Deep-ECG, designed for the robust detection of Atrial Fibrillation (AF) utilizing three publicly available single-lead ECG databases from PhysioNet (CPSC2018, Georgia, and ECG-Arrhythmia). The proposed system consists of an innovative dual-branch Transformer architecture that accurately classifies atrial fibrillation against normal sinus rhythms.

The computational efficiency and high diagnostic sensitivity of the proposed system demonstrate its strong suitability for real-time and outpatient atrial fibrillation screening applications.

Keywords: Electrocardiogram, Atrial Fibrillation, Classification, Deep Learning, Transformer.

Résumé

L'électrocardiographie (ECG) reste la méthode standard pour identifier les irrégularités cardiaques. La détection automatique des anomalies à partir de l'ECG peut aider à la détection précoce, au diagnostic et à la prévention des maladies cardiovasculaires. Cette thèse présente un nouveau système d'apprentissage profond, nommé Deep-ECG, conçu pour la détection robuste de la fibrillation auriculaire (FA) en utilisant trois bases de données d'ECG à dérivation unique publiquement disponibles issues de PhysioNet (CPSC2018, Georgia et ECG-Arrhythmia). Le système proposé repose sur une architecture Transformer innovante à deux branches qui classe avec précision la fibrillation auriculaire par rapport aux rythmes sinusiens normaux.

L'efficacité calculatoire et la haute sensibilité diagnostique du système proposé démontrent sa forte adéquation pour les applications de dépistage de la fibrillation auriculaire en temps réel et en ambulatoire.

Mots-clés : Électrocardiogramme, Fibrillation auriculaire, Classification, apprentissage profond, Transformer.

ملخص

يظل تخطيط كهربية القلب الوسيلة القياسية لتحديد اختلالات وظائف القلب، حيث يمكن للكشف الآلي عن الاضطرابات عبر تخطيط كهربية القلب أن يساعد في الكشف المبكر، والتشخيص، والوقاية من أمراض القلب والأوعية الدموية. تقدم هذه الأطروحة نظامًا جديدًا للتعلم العميق، يُعرف بـ "**نظام تخطيط القلب العميق**"، مصمم للكشف الفعال عن الرجفان الأذيني بالاعتماد على ثلاث قواعد بيانات عامة لتخطيط كهربية القلب أحادية القناة مستمدة من منصة "فيزيونت" وهي: (سي بي إس سي 2018، وجورجيا، وقاعدة بيانات اضطرابات نظم القلب). يتكون النظام المقترح من بنية محوّل مبتكرة ثنائية الفروع تصنف بدقة الرجفان الأذيني مقابل نظم الجيوب الأنفية الطبيعي.

تثبت الكفاءة الحسابية والحساسية التشخيصية العالية للنظام المقترح ملاءمته القوية لتطبيقات فحص الرجفان الأذيني في الوقت الفعلي وفي العيادات الخارجية.

الكلمات المفتاحية: تخطيط كهربية القلب، الرجفان الأذيني، التصنيف، التعلم العميق، المحوّل.

CONTENTS

List of Figures VI

List of Tables VIII

Nomenclature IX

General Introduction 1

Problematic 2

Motivation and Objectives 3

Thesis Structure 3

1 Deep learning 5

1.1 Introduction 5

1.2 Artificial Intelligence (AI) 5

1.3 Machine Learning (ML) 7

1.4 Deep Learning (DL) 9

1.5 Convolutional Neural Networks (CNNs) 9

1.5.1 Convolutional Layer 10

1.5.2 Pooling Layer 11

1.5.3 Fully Connected Layer 11

1.6 Long Short-Term Memory Networks (LSTMs) 12

1.7 Transformers and Attention Mechanisms 13
1.7.1 Scaled Dot-Product Attention 14
1.7.2 Multi-Head Attention 14
1.7.3 Positional Encoding 15
1.8 Conclusion 16

2 ECG Overview **17**
2.1 Introduction 17
2.2 The Electrocardiogram (ECG) 18
2.3 ECG Waveform Parameters 19
2.4 ECG Classification Challenges 22
2.5 ECG and Atrial Fibrillation (AF) 23
2.6 Related Work 25
2.7 Conclusion 29

3 Design Of Deep-ECG System **31**
3.1 Introduction 31
3.2 Suggested Approach 31
3.3 System Development Phases 33
3.3.1 Preprocessing 33
3.3.2 R-Peak Detection 35
3.3.3 Branch 1: Beat Morphology 36
3.3.4 Branch 2: RR Rhythm Features 41
3.3.5 Cross-Attention Fusion 43
3.3.6 Classifier 44
3.4 Conclusion 45

4 Results And Discussion **47**
4.1 Introduction 47
4.2 Development Environment 48
4.2.1 Hardware Specifications 48

4.2.2 Python Modules 48
4.3 Datasets 49
4.4 Implementation and Results 53
4.4.1 Experimental Setup 53
4.4.2 Evaluation Metrics 54
4.4.3 Performance of Deep-ECG system 56
4.4.4 Comparative Analysis of Implemented Architectures 58
4.4.5 Comparison of Proposed Model with Related Works 59
4.4.6 Explainability 62
4.4.7 Web Application – Deep-ECG Platform 63
4.5 Conclusion 67

General Conclusion **69**

LIST OF FIGURES

1.1 Artificial Intelligence Components 6
1.2 Illustration of the machine learning approaches 8
1.3 CNN Architecture 10

2.1 Normal ECG waveform showing the P, Q, R, S, and T waves with annotated components. 21
2.2 Anatomical and electrical comparison between a Normal Sinus Rhythm and Atrial Fibrillation, showing the chaotic electrical signals in the atria. 24
2.3 A 2lead ECG tracing illustrating the primary clinical characteristics of Atrial Fibrillation, including the absence of sinus P waves,and highly irregular RR intervals. 25

3.1 Proposed Deep-ECG System for Atrial Fibrillation Detection. 33
3.2 ECG preprocessing comparison: raw (left) and preprocessed (right) signals for Normal Sinus Rhythm (top) and Atrial Fibrillation (bottom). 35
3.3 R-Peak Detection architecture overview. 36
3.4 Architecture of the Beat CNN. 37
3.5 Branch 1 (Beat Morphology)architecture 40

3.6 Branch 2 (RR Rhythm) architecture 43
3.7 Cross-Attention Fusion and Classifier architecture . . . 45

4.1 Summary of the three ECG data sources after filtering. . 51
4.2 Combined dataset statistics and class balance. 52
4.3 Stratified train/validation/test split. 53
4.4 Training and validation curves for the Deep-ECG Model: loss (left), AUC (centre), and accuracy (right). 56
4.5 Confusion matrix (left) and ROC curve (right) for DEEP-ECG 57
4.6 ECG visualisation strip showing 11 beats with P-wave colour annotations (green = present, red = absent) and the selected best beat highlighted by an orange dashed frame. 63
4.7 Dashboard overview displaying session statistics (total analyses, AF detected, Normal), pipeline description, and navigation controls. 65
4.8 Interactive ECG Analysis interface displaying the input and signal upload panel, the processed Lead II ECG waveform with automated segment annotations, and the real-time AI classification result sidebar with medical explanation metrics. 66
4.9 Patient Archive panel showing a chronological log of ECG analyses with AF probabilities, classification decisions, and report/delete actions. 66
4.10 Comprehensive AI-generated clinical report interface within the Last Report module, illustrating patient diagnostics, automated parameter explanations, and browser-ready printing options. 67

LIST OF TABLES

2.1 Key ECG Parameters and Their Clinical Significance . . 21
2.2 Quantitative Comparison of Deep Learning Models for AF Detection 29

4.1 Summary of shared training hyperparameters for all proposed models. 54
4.2 Detailed performance metrics of Deep-ECG on the test set. 56
4.3 Comprehensive performance comparison of the implemented models on the test set. 58
4.4 Error distribution comparison across implemented models. 59
4.5 Quantitative Comparison of Deep Learning Models for AF Detection 60

NOMENCLATURE

Abbreviations / Acronyms

AF	Atrial Fibrillation
AI	Artificial Intelligence
AMP	Automatic Mixed Precision
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
AV	Atrioventricular
BCE	Binary Cross-Entropy
BM	Beat Morphology
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
DL	Deep Learning
ECG	Electrocardiogram
FFN	Feed-Forward Network
FN	False Negatives
FP	False Positives

HR	Heart Rate
HRV	Heart Rate Variability
LSTM	Long Short-Term Memory
MHA	Multi-Head Attention
ML	Machine Learning
MLP	Multi-Layer Perceptron
NSR	Normal Sinus Rhythm
RNN	Recurrent Neural Network
SA	Sinoatrial
TN	True Negatives
TP	True Positives

GENERAL INTRODUCTION

The Electrocardiogram (ECG) has been recognized for a long time as the main non-invasive tool for recording the bioelectric signals of the heart. Thus, an ECG can allow the determination of many important physiologic factors relating to the heart, including; heart rate, regularity of rhythm, and morphology (shape) of waveforms (i.e. P-waves, QRS complex and RR intervals). The importance of continuous ECG monitoring to detect complex arrhythmias such as Atrial Fibrillation (AF) is essential. Manual interpretation of these signals can be very laborious, especially when considering long periods of time (i.e., 24 to 48 hour Holter recordings). Interpretation requires specialized clinical skills and may also be subject to a number of additional issues such as fatigue of the observer and an overwhelming amount of data generated from long-term recording. Therefore, creating reliable automatic analysis systems for EC during both routine and emergency diagnosis is imperative to support cardiology providers. While many deep learning techniques have improved in their ability to detect and identify heart conditions such as atrial fibrillation (AF), one critical limitation of most current methods is the fact that they rely almost exclusively on a single domain of ECG features (i.e., morphology of individual ECG beats or temporal variations in RR intervals). To accurately detect conditions such as

AF, both types of feature must be considered together: morphological changes (such as the absence of clearly identifiable P-waves and the presence of fibrillatory waves) and rhythmic changes (R-R intervals that are highly variable). To overcome this limitation, we propose a new, dual-branch model, called Deep-ECG. One of the defining characteristics of the Deep-ECG model is that it will use a cross-attentional fusing mechanism, allowing it to extract and combine morphological and rhythm features simultaneously, ultimately producing a more complete and clinically useful classification of AF for ECG signals recorded with a single lead.

Problematic

Automated ECG-based methods for detecting atrial fibrillation (AF) face several technical challenges.

For example, ECGs can be negatively affected by various types of noise in practice, all of which degrade signal quality and complicate feature extraction. In addition to noise, ECG patterns can vary considerably between individuals, making it difficult to distinguish cardiac activity recordings when viewing the ECGs of two different patients. Consequently, the model cannot be generalized to individuals who have not been seen before.

Furthermore, the general datasets used to train AF detection models suffer from an unbalanced distribution of classes, which can lead to bias during model training, as there are significantly more normal sinuses than cases with atrial fibrillation.

Finally, accurate AF detection requires analyzing the morphological characteristics of individual impulses, as well as their long-term arrhythmias, by studying the interval variability (RR). Developing a robust model that combines these two types of information is one of the biggest

challenges facing researchers today.

Motivation and Objectives

- Despite being one of the most common causes of heart disease, atrial fibrillation is often misdiagnosed.
- Its intermittent nature makes diagnosis difficult, and manual interpretation of electrocardiograms (ECGs) is limited.
- This was the impetus behind the development of the Deep-ECG system, which aims to automatically detect atrial fibrillation and other cardiac arrhythmias by analyzing ECG signals in a manner similar to a cardiologist,
- While addressing real-world problems such as noise, data imbalances between different patient groups, and others.

Thesis Structure

This work is structured into four chapters.

- **Deep Learning:**
  This chapter explains the theoretical framework of Machine Learning and Deep Learning, with details on the principal networks of neural architectures used in this research.
- **ECG Overview:**
  This chapter discusses the medical background of ECG signals and Atrial Fibrillation, and provides an overview of the literature regarding automated classification of ECG signals.

- **Design Of Deep-ECG System:**
  This chapter presents the proposed Deep-ECG system in terms of its overall architecture, data processing pipeline, and model training.

- **Results & Discussion:**
  In this chapter, we discuss the technical aspects of our project, such as the development environment, the datasets used, and the results obtained.

CHAPTER 1

DEEP LEARNING

1.1 Introduction

This chapter introduces the basic theoretical concepts to understand sophisticated computational models. The chapter starts with an overview of the basic principles of Artificial Intelligence (AI) and Machine Learning (ML), followed by an elaboration on the basic concepts of Deep Learning (DL). The chapter then introduces some of the most popular neural networks, with a focus on Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformers, to explain how these models identify complex patterns in time-series data.

1.2 Artificial Intelligence (AI)

The domain of Artificial Intelligence (AI) focuses on developing systems that are capable of executing tasks that are normally associated with human intelligence. These systems are designed to learn from experience, recognize intricate patterns, and solve problems independently. The fundamental aim of AI research is to create models that are capa-

ble of perceiving their environment, understanding the information that they are receiving, and making independent decisions.[1]

Artificial Intelligence (AI) is a highly transformative technology that is increasingly being embedded into all aspects of the healthcare industry. Cardiologists normally rely on their expertise to interpret intricate physiological patterns, such as an electrocardiogram (ECG) trace. Nevertheless, AI-based systems have been developed to automate this process[1]. While AI refers to the general concept of machine intelligence, the most powerful manifestation of this concept can be achieved by restricting it to specific domains such as machine learning and deep learning, as depicted in Figure 1.1.

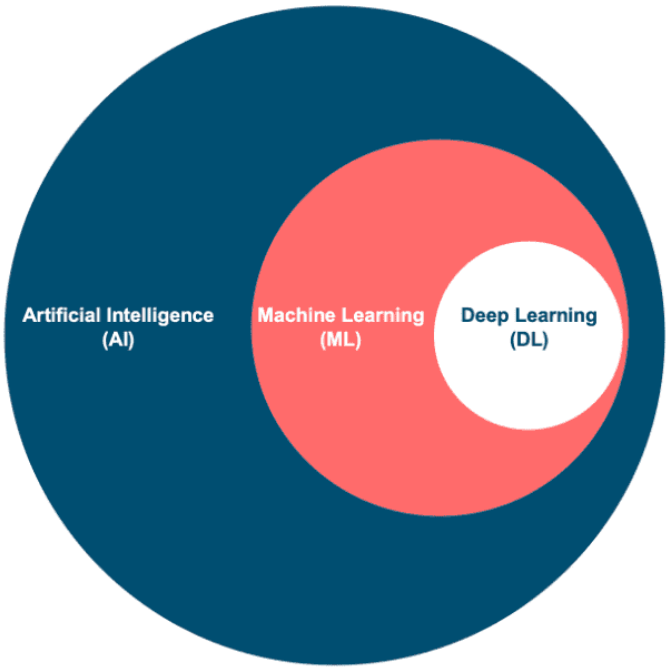


Figure 1.1: Artificial Intelligence Components

1.3 Machine Learning (ML)

Machine Learning (ML) acts as an integral part of Artificial Intelligence. The major objective of this field is to create systems that can learn from data and utilize this knowledge to make predictions. Arthur Samuel popularized the term "Machine Learning" to define a field that helps computers learn from data without any direct programming. The concept was fundamentally shaped by Arthur Samuel in 1959, who described it as a field of study that focuses on developing computers able to learn from experience without the need for detailed and explicit programming [3].

ML algorithms create a mathematical model to detect data patterns based on training data that includes input data and corresponding output data in a supervised manner. Tom M. Mitchell defined ML in a more formal way, which is most commonly used by individuals when they define ML as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E[2].

The Machine Learning algorithms are divided into three major paradigms based on the nature of the data and the learning process, as shown in Figure 1.2.

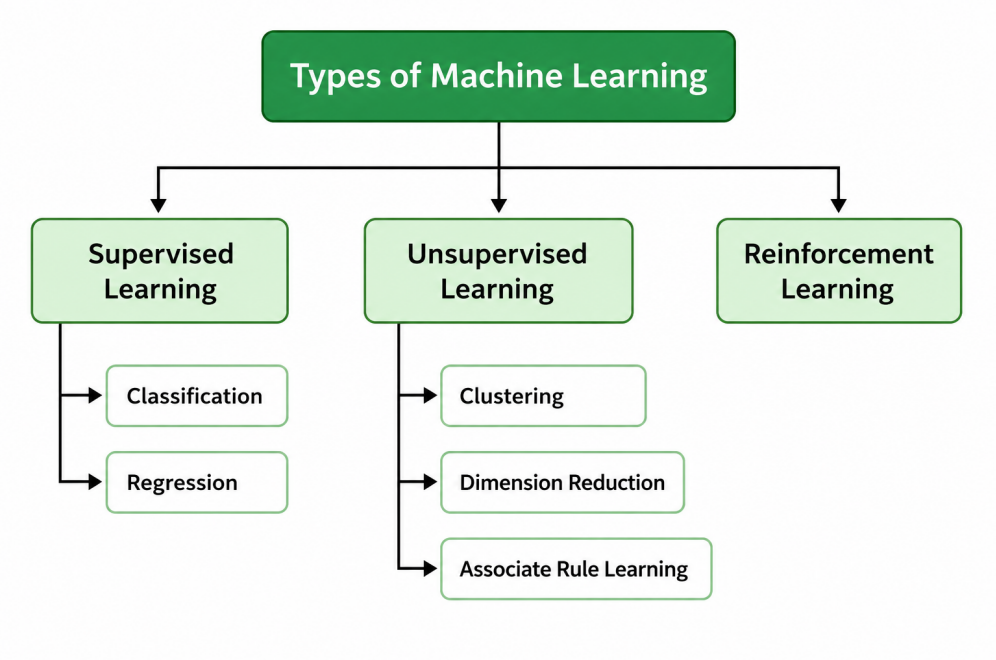


Figure 1.2: Illustration of the machine learning approaches

- **Supervised Learning:** This approach involves a training process that uses labeled data sets that consist of input elements along with their respective output values. A function is created that maps input elements to their respective output values, which enables the algorithm to make predictions on unknown data sets[2].
- **Unsupervised Learning:** This approach involves an algorithm that processes data sets without any output values. The main aim is to allow the algorithm to automatically search for hidden patterns, basic patterns, and clusters within the input data sets[2].
- **Reinforcement Learning:** This approach involves an algorithm that learns to make a series of decisions using an environment. The algorithm has to find a way to make decisions that will bring maximum rewards since no instructions are given on how to make decisions[2].

1.4 Deep Learning (DL)

Deep Learning (DL), on the other hand, is a powerful category of Machine Learning that uses deep neural networks composed of multiple artificial neural network layers to train systems about data based on unprocessed input[4]. Deep Learning systems are capable of recognizing increasingly complex and advanced patterns based on their multiple levels of processing[4].

Deep Learning attains its best performance potential based on its automatic feature learning system that does not rely on the development of features to analyze data that is present in high-dimensional spaces and unstructured forms such as images, audio, text, and time series. The complex computing architectures learn representations of data that contain different levels of abstraction using their various levels of processing to develop highly accurate and dependable predictive models in different complex domains.

1.5 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a specific type of multi-layered artificial neural network[5]. The specific design of this type of artificial neural network was first described by LeCun et al[6]. The design is now a key component in the implementation of deep learning. The design is based on a hierarchical model where the initial processing layers are primarily used to identify basic features, while the later processing layers are used to effectively combine the basic features to form high-level attributes of the input [5].

CNNs provide higher accuracy and improve the overall performance of a system significantly, mainly due to the two specific architectural designs of this type of artificial neural network, which are local connectivity

and shared weights[5]. The specific design effectively eliminates the dimensionality problem related to the extracted features by the neural network, thus reducing the computational complexity [5]. To effectively deliver high performance, the basic design of a Convolutional Neural Network consists of a sequence of specific layers, as shown in Figure 1.3. The sequence consists mainly of a convolutional layer, a pooling layer, and a fully connected layer.

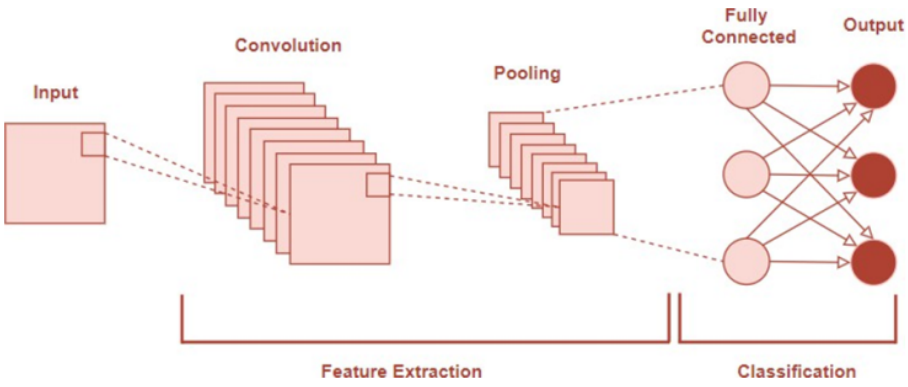


Figure 1.3: CNN Architecture

1.5.1 Convolutional Layer

The convolutional layer serves as the core building block of a CNN. It uses mathematical convolution to extract features from input data. The layer uses learnable filters, which also function as kernels, to slide across input data and identify particular local patterns. The discrete convolution operation combines the input image I with the kernel K to create the feature map S through its mathematical definition:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{1.1}$$

The spatial indices of the process are represented by i and j while m and n define the dimensions of the kernel . The role of the layer is carried out through local interconnection[s between constituent elements,

in which shared weights are used to obtain feature maps that require fewer parameters to be learned [5].The output obtained from the linear convolution operation is then subjected to a non-linear activation function, which is the Rectified Linear Unit (ReLU) given by the equation $f(x) = \max(0, x)$[6].

1.5.2 Pooling Layer

After the convolutional operation, a pooling or sub-sampling layer is often used in the network, which gradually reduces the spatial size of the extracted features. The network experiences decreased complexity through down-sampling, which maintains essential information, thus controlling overfitting according to[5].

The most commonly utilized approach is Max Pooling, which extracts the maximum value from a defined local region $P_{i,j}$. This operation can be expressed as:

$$y_{i,j} = \max_{(m,n)\in P_{i,j}} x_{m,n} \tag{1.2}$$

Where $x_{m,n}$ represents the activation values within the pooling region. The system remains stable in operation due to its capacity to accommodate minor variations.

1.5.3 Fully Connected Layer

The last part of a CNN is a set of fully connected layers, which is equivalent to a multilayer perceptron. After going through several convolutional and pooling layers, the CNN finally produces feature maps of varying dimensions. The feature maps are then flattened into a single dimension. The main purpose of this last layer is to interpret the features and produce the output. The operation can be explained as follows:

$$y = f(Wx + b) \tag{1.3}$$

The flattened input vector x goes through a computation where it gets multiplied by the learned weight matrix W and added with the bias vector b. Then, a non-linear activation function f is applied, which includes the Softmax step for the final prediction probabilities.

1.6 Long Short-Term Memory Networks (LSTMs)

Although Convolutional Neural Networks are highly potent for feature learning, Recurrent Neural Networks (RNNs) are traditionally used for modeling sequential data. However, conventional RNNs greatly suffer from the vanishing gradient problem, which is computationally expensive for learning dependencies in long sequences. In order to overcome this critical limitation of conventional RNNs, Long Short-Term Memory (LSTM) networks were proposed by Hochreiter and Schmidhuber[7].

An LSTM network makes use of a special memory unit, which is called a cell, and can maintain the state of the cell for a long period of time. The flow of information in and out of the cell is carefully controlled by three different controlling structures, which are collectively called gates[5]. These gates use a sigmoid neural net layer and a pointwise multiplication operation.

The equations of computation with respect to individual recurrent gates for a given LSTM cell at a specific t time step are as follows:

- **The Forget Gate:** This gate enables the selection of information that needs to be removed from the previous cell state C_{t-1}. The system takes in both the previous hidden state h_{t-1} and the current input x_t to produce an output value that ranges from 0 to 1:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1.4}$$

- **Input Gate and Candidate Memory:** The input gate decides what new information will be securely stored in the cell state. A

sigmoid layer defines the specific values to update (i_t), and a tanh layer generates a vector of new candidate values ($\tilde{C}_t$):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1.5}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{1.6}$$

- **Cell State Update:** The old cell state C_{t-1} is then computationally updated to a new state C_t. The process starts with the system discarding information from the forget gate before it proceeds to incorporate new candidate values:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{1.7}$$

- **Output Gate:** The gate manages output information based on newly filtered cell state information. The system produces a controlled output that shows cell state information through the new hidden state h_t:

$$h_t = o_t * \tanh(C_t) \tag{1.8}$$

The equation displays the learned weight matrices W and bias vectors b which are used in the gate operation together with the logistic sigmoid function σ and the element-wise multiplication operator $*$ for computation.

1.7 Transformers and Attention Mechanisms

Even though the LSTM networks are able to effectively handle the vanishing gradient issue for sequential data, the very nature of the sequential processing limits the parallelization capability of the network, resulting in increased computational time for long sequences. In order to effectively handle the sequential data, a new architecture called the Transformer was proposed by Vaswani et al.[8]. The proposed architecture

uses Attention mechanism to create global input-to-output connections which remove the need for recurring and convolutional operations. The complete computational strength of the Transformer system comes from its various mathematical components which deliver multiple processing functions.

1.7.1 Scaled Dot-Product Attention

The basic block of the Transformer architecture is the "Scaled Dot-Product Attention". The attention process works on three key vectors from the input data: Queries, Keys, and Values, denoted as Q, K, and V, respectively. The dot product of Q and K, normalization by dividing by the square root of the key dimension, and finally applying the softmax function are used in the process. The mathematical expression for the process can be given as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.9}$$

This mathematical expression enables the model to dynamically allocate different weights for different components of the input sequence, regardless of the distance between them in the sequence.

1.7.2 Multi-Head Attention

Instead of using a single attention operation, the Transformer uses a variant known as Multi-Head Attention. This means that the queries, keys, and values are linearly transformed h times using different linear transformations. The attention operation is applied in parallel to each of the linear transformations. This means that h output vectors are produced. These output vectors are combined and another linear transformation is applied. This can be represented as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{1.10}$$

Where each individual head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{1.11}$$

In this respect, the projections are defined by the parameter matrices W_i^Q, W_i^K, W_i^V, and W^O. The Multi-Head Attention mechanism allows the model to attend to information from multiple subspaces in different positions.

1.7.3 Positional Encoding

As the Transformer architecture does not rely on recurrence and convolution, it does not have an inherent understanding of the ordered nature of sequences. In order for the model to leverage this information within the ordered sequence, positional information is added to the input embeddings at the base of the encoder and decoder stacks. The positional encodings make use of sine and cosine functions with different frequencies:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{1.12}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{1.13}$$

Where pos represents the position of the data point in the sequence, i represents the dimension, and d_{model} corresponds to the embedding dimension. This mathematical addition helps to ensure that the architecture retains the essential time context of the processed signals.

1.8 Conclusion

This chapter has provided the underlying theoretical foundation that is necessary for understanding the advanced computational models used in modern signal processing techniques. The chapter started with an overview of Artificial Intelligence and Machine Learning, followed by a discussion of the instrumental capabilities of Deep Learning techniques. In particular, this chapter discussed the architectural mechanics of Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for sequential data, and the state-of-the-art Transformer model for parallelized sequence data.

CHAPTER 2
ECG OVERVIEW

2.1 Introduction

Cardiovascular disorders still constitute the world's leading cause of death, responsible for about 19.8 million deaths each year [9]. Within modern medical practice, it is important to diagnose cardiac conditions accurately at an early stage. The ECG serves as a primary tool for diagnosing cardiac disorders without causing any harm and at a relatively low cost. Of the many forms of cardiac dysrhythmia, AF is one of the more common, as it increases the risk of stroke and heart failure [13].

Continuous ECG interpretation is an extremely tedious and difficult procedure that involves the skills of cardiologists who have expertise. Manual interpretation is prone to errors owing to the human element, such as fatigue, observer variations, and morphological changes occurring due to some pathologies.

The past few decades have seen the introduction of automated ECG analysis systems based on artificial intelligence (AI), which have emerged as highly effective tools to assist clinicians in diagnosis. While machine learning laid down the groundwork for ECG analysis by extracting features manually, deep learning has revolutionized this process through its

ability to learn features hierarchically.

In this chapter, concepts from both the medical field and computation are considered. The chapter starts by discussing the physiology behind the creation of an Electrocardiogram (ECG) signal, the different components present in an ECG signal, and common diagnostic features used. Thereafter, the chapter discusses some of the inherent difficulties that are faced during automated ECG classification. Lastly, a thorough overview of machine learning and deep learning approaches to ECG analysis is presented, highlighting some of the most common challenges facing researchers today.

2.2 The Electrocardiogram (ECG)

The human heart acts as a sophisticated electromechanical device, whereby each mechanical contraction is triggered by an electrical depolarization process. Electrocardiography (ECG) is a continuous and non-invasive method for recording the heart's bioelectric activity through electrodes attached to the body's surface [12]. Electrical activation begins in the SA node, which acts as the natural pacemaker in the heart and then propagates through the AV node, the Bundle of His, and the Purkinje fibers, leading to ventricular contraction. Through the continuous recording of the tiny voltage changes over time, ECG plays a vital role in providing clinicians with information about the heart's electrophysiological condition; hence, making it an essential tool for diagnosing various cardiac conditions.

These projections of electrical vectors depend on the orientation of the body axes. To capture spatial variations, the electrocardiogram is recorded using multiple electrode placements called "leads."

- **Standard 12-Lead ECG:** This technique remains the gold standard in clinical practice. It involves ten electrodes that are attached to the body's extremities and the chest area, producing twelve different electrical views of the heart. The 12-lead system provides comprehensive spatial information about cardiac electrical activity which aids clinicians in both diagnosing heart problems and identifying regions of myocardial ischemia.[10]

- **Single-Lead and Wearable ECG:** While the 12-lead system provides complete spatial data, it is too cumbersome for long-term continuous analysis. Single-lead or reduced-lead systems, such as those used in Holter monitoring and consumer devices (e.g., smartwatches), provide continuous recordings. These monitors can record from one or two leads over extended periods (up to 24-48 hours and more). Such analyses are highly effective in capturing transient arrhythmias that might be missed in short clinic visits. The availability of large collections of such data sets from the general public, specifically the PhysioNet database, has proven invaluable in the testing and evaluation of automated arrhythmia detectors[11].

2.3 ECG Waveform Parameters

A normal ECG trace for one heartbeat usually contains several waveforms, each representing an electrophysiological process in the heart[10]. Morphology is defined by the size, time span, and orientation of each of these waveforms; the main criteria for diagnosis include all three aspects.

- **P wave:** The first positive deflection represents depolarization of the atria, which leads to the contraction of the atria. In most cases, this wave typically has a small amplitude.

- **PR interval:** This time period measures from the start of the P wave to the beginning of the QRS complex, indicating conduction from the atria to the ventricles, primarily through the AV node.
- **QRS complex:** The dominant and largest wave of an ECG signal indicates ventricular depolarization. It consists of a small negative Q wave, positive R wave, and negative S wave. Because of its unique shape and large magnitude, the R wave can be used as a point of reference when dividing heartbeats and calculating heart rate [12] .
- **T wave:** The T wave signifies ventricular repolarization and manifests as a positive deflection following the QRS complex.
- **QT interval:** The duration of this interval starts at the beginning of the QRS complex and ends at the end of the T wave.

Aside from these waves and intervals, the ST segment is another important segment as it runs from the end of the S wave to the beginning of the T wave and has clinical significance when it deviates from the isoelectric line. A typical ECG waveform with its main components is illustrated in Figure 2.1.

The waveform characteristics of atrial fibrillation become irregular when the P waves disappear and the RR intervals begin to show unpredictable changes [10].

Table 2.1: Key ECG Parameters and Their Clinical Significance

Parameter	**Description**	**Normal Range**	**Clinical Relevance**
Heart Rate (HR)	Number of beats per minute	60–100 bpm	Bradycardia / Tachycardia
RR Interval	Time between two consecutive R peaks	0.6–1.2 s	Arrhythmia detection
PR Interval	Time from start of P wave to start of QRS	120–200 ms	AV conduction blocks
QRS Duration	Width of the QRS complex	$<$ 120 ms	Bundle branch block
QT Interval	Time from start of QRS to end of T wave	350–440 ms	Long QT syndrome
ST Segment	Segment between S and T waves	Isoelectric	Myocardial ischemia

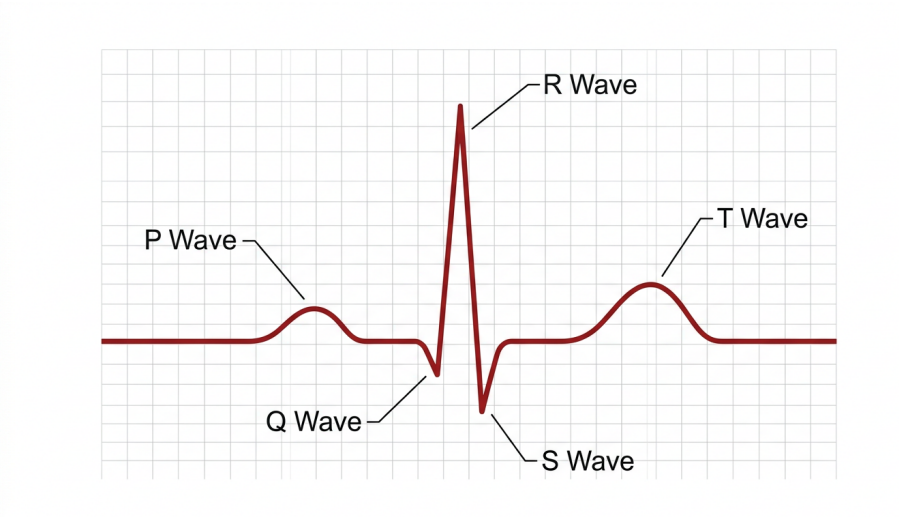


Figure 2.1: Normal ECG waveform showing the P, Q, R, S, and T waves with annotated components.

2.4 ECG Classification Challenges

Although considerable progress has been made in the field of machine learning and artificial intelligence in medical diagnostics, the problem of classifying ECGs accurately and reliably in real-life situations still poses a serious challenge. Atrial fibrillation detection faces major obstacles because irregular heart rhythms combined with different patterns seen in each patient create problems for precise diagnosis. According to recent comprehensive surveys [12], several inherent difficulties and domain-specific challenges persist in limiting the effectiveness and applicability of automatic ECG interpretation systems:

- **Noise and Artifacts:** Because of electrode contact issues and powerline interference from the electrical grid, raw ECG signals are often affected by outside forces. Previous studies have shown that noise from physical muscle movement and motion artifacts changes the shape of the waveform, which leads the system to provide incorrect identification results [12]. Since this noise is quite strong, raw ECG signals are frequently damaged during the recording process.
- **Inter-Patient Variability:** Differences in age, gender, and body structure cause ECG morphology to change among different individuals. Medical professionals observe that heart-related health issues also make ECG morphology vary from person to person. Because a group of human subjects is used to train a model, poor generalization is often shown by the model when new individuals are tested. ECG morphology remains a difficult challenge for these models because every human being has a unique heart signature.
- **Class Imbalance:** Within medical data collections, normal sinus rhythm is found much more frequently than arrhythmias which are dangerous yet uncommon. Because this disproportion exists, iden-

tification tools favor common patterns, which weakens the capacity for finding life-threatening arrhythmias. Previous studies have shown that the heavy presence of normal data creates a weight against rare cases during the learning process.

- **Dependency on R-Peak Detection:** Medical professionals believe that many old and early complex computer methods depend on finding the exact R-peak to carry out heartbeat division into individual parts. If a mistake happens during this early preparation stage, the mistake moves through the whole sequence of sorting heartbeats. When recordings have much noise, the problems created by early preparation mistakes become very large. Accuracy is often lost because the whole sequence is affected by these initial errors.

- **Long-Range Temporal Dependencies:** Correct identification for various arrhythmias needs a clear view of heart patterns across long chains of heartbeats. Long strings of heartbeats must be analyzed by machines so that the specific rhythm becomes clear. Even though RNNs and LSTMs try to learn from the order of data, these models frequently fail when sequences become very long. It has been observed that these models face difficulty because of vanishing gradients and speed problems in calculation.

2.5 ECG and Atrial Fibrillation (AF)

Atrial fibrillation (AF) has been found to be the most common sustained cardiac arrhythmia characterized by rapid and irregular electrical activity in the atrium [12]. As illustrated in Figure 2.2, it impairs the efficient atrial contractions to result in an irregular ventricular response. Its clinical relevance has been associated with severe complications, such

as stroke, heart failure, and high mortality rates. Therefore, prompt and accurate AF diagnosis becomes imperative.

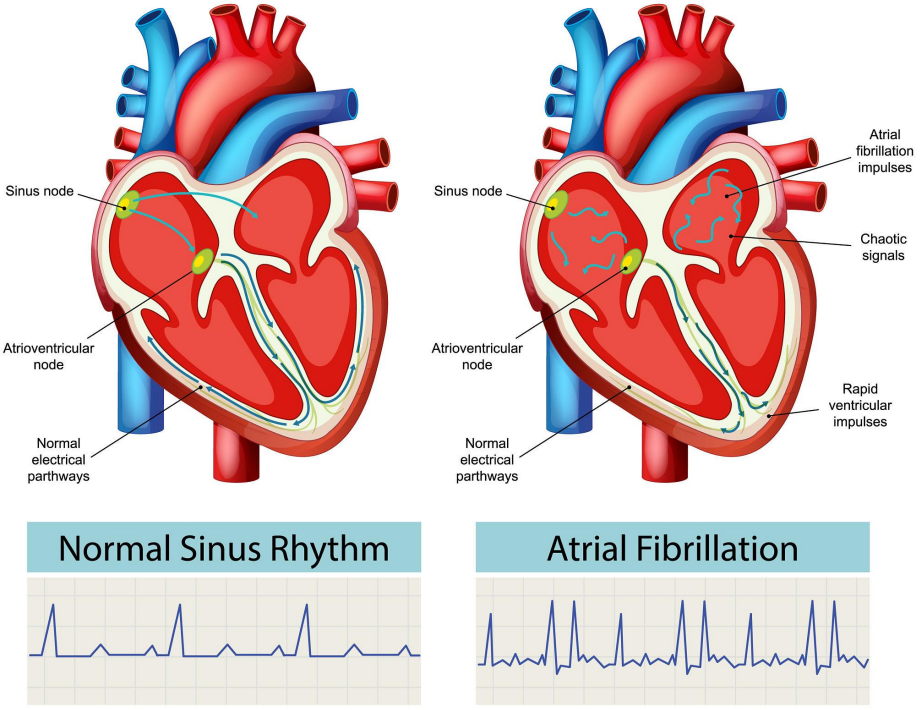


Figure 2.2: Anatomical and electrical comparison between a Normal Sinus Rhythm and Atrial Fibrillation, showing the chaotic electrical signals in the atria.

Based on the electrocardiographic view, AF presents itself through certain features that differentiate it from a normal sinus rhythm, as shown in Figure 2.3. One of the key characteristics of AF includes absent clear P waves and their replacement with irregular fibrillatory waves. Furthermore, irregularity in the RR interval becomes evident due to chaotic transmission of electrical impulses into the ventricles. Such irregularities make AF detection especially hard, more so with long-term ECG data.

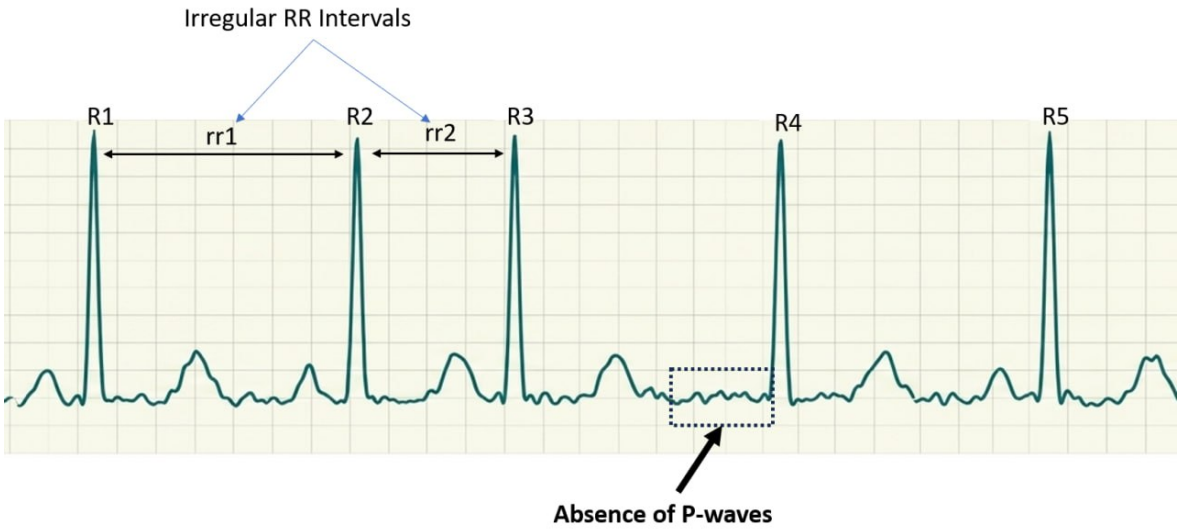


Figure 2.3: A 2lead ECG tracing illustrating the primary clinical characteristics of Atrial Fibrillation, including the absence of sinus P waves,and highly irregular RR intervals.

To address this, deep learning models have been used to develop classifiers to detect AF accurately. Such a decision becomes necessary since AF diagnosis entails morphological and long-range temporal information extraction.

2.6 Related Work

Healthcare has greatly benefited because of advancements in deep learning that has remarkably improved automated ECG analysis systems' performances. Researchers have dedicated their studies to atrial fibrillation (AF) detection because of its important role in clinical practice according to recent research findings [13]. Unlike traditional machine learning approaches that are trained using hand-tuned features, state-of-the-art models can learn deep, discriminative representations directly from raw ECG time-series signals.

The early studies proved that CNNs can learn discriminative features from ECG signals by analyzing them as input in their raw form. Hannun

et al[14]. developed a deep CNN architecture with 34 layers designed for detecting arrhythmia in single-lead ECG signal recordings at the level of an experienced cardiologist. The CNN was trained on a large proprietary Zio Patch database including 29,163 patients and 14 different rhythm classes. While the main scope of the study included several rhythm classes as opposed to AF specifically, AF represented one of the target rhythm classes. The achieved F1-scores at sequence and set levels amounted to 0.604 and 0.667 correspondingly for AF classification task.

To better capture the temporal dynamics among the ECG signal, there are several versions of hybrid CNN-recurrent models that have been created. For example, Ping et al[17]. developed an 8 layer CNN (Convolution Neural Network) which has two layers of shortcut connections and 1 layer of LSTM (Long Short Term Memory), called 8CSL. Here the CNN is responsible for extracting higher level ECG feature extraction while the LSTM is used to identify the relationships at each point in time within the ECG signal. Shortcut connections were added to the original design to speed up information transfer and increase communication through out training. Results from testing this model against the Computing in Cardiology Challenge 2017 dataset yielded average test F1 scores of 84.89%,89.55%, and 85.64% when processing ECG signals that had lengths of 5 seconds, 10 seconds, and 20 seconds respectively.

As the use of attention mechanisms gains momentum, Hu et al[16]. created ECG DETR, a new deep neural network based on transformers that represents atrial fibrillation detection as a type of object detection through continuous single-lead segments of ECG. They combine 1D CNN with a transformer encoder-decoder architecture, removing the need for specific heartbeat segmentation when modeling, and allowing the model to model the long-distance interdependence between heartbeats using self-attention. Their model achieved state-of-the-art results for the de-

tection of atrial fibrillation in ECG segments by taking advantage of the long-duration episodes that characterise atrial fibrillation. the model (specifically the ECG DETR α variant) yielded an outstanding overall accuracy and F1-score of 99.23

Other approaches involve transformation of one-dimensional ECG signals to two-dimensional time-frequency representation, which makes it possible to use image processing CNN models. The work by Wei et al[15]. suggested an automatic method of detecting atrial fibrillation using ECG data, where ECG preprocessing, generating spectrograms, and transferring learned knowledge from an efficient fine-tuned model based on EfficientNet B0 was involved. For this purpose, the researchers used PhysioNet/Computing in Cardiology Challenge 2017 dataset for binary classification of normal rhythm versus AF cases. The number of examples in the dataset was 5,154 for normal and 771 for AF cases; the model showed 88.2% of F1-score and 97.3% of accuracy.

However, more recently, wavelet approaches have been utilized for the purpose of extracting better time-frequency characteristics from a single lead ECG. In one such study, Xie et al[18]. proposed the CWT-MB-ResNet for automatic detection of atrial fibrillation (AF). This approach uses a continuous wavelet transform to transform the ECG readings into two-dimensional scalograms which are then classified using the ResNet-18 algorithm with the help of a multi-branch structure. The objective behind using the multi-branch structure is to address the problem of class imbalance as normal ECG examples usually outnumber those of AFs.The proposed CWT-MB-ResNet model demonstrated robust prediction performance, achieving an F1 score of 0.8865 on the PhysioNet dataset, as well as an F1 score of 0.7396 on the OUHSC dataset.

Another trending development in this domain is that apart from the task of segment level AF classification, it also attempts beat level

AF localization. Recently, Wang et al[19] proposed AF-DETR that uses DETR-like object detection to localize and classify AF heartbeats. Rather than classifying the whole segments of an ECG as done in most existing studies, AF-DETR tries to find out the location and class of heartbeats in the ECG data. This is accomplished using CNN backbone with Transformer encoder-decoder architecture and two-dimensional bounding box around heartbeats. This model was tested on some publicly available ECG databases like CPSC2021, AFDB, LTAFDB, MITDB, and NSRDB. The beat level F1 scores obtained were 96.77% for AFDB and 96.20% for LTAFDB, while the segment level AF F1-scores were 97.67% for AFDB and 97.72% for LTAFDB.

Overall, the literature demonstrates a clear evolution from traditional ML algorithms to deep learning models, and more specifically to hybrid approaches that make the best of both convolutional and attention mechanisms. A quantitative comparison of these representative state-of-the-art methodologies is presented in Table 4.5.

Table 2.2: Quantitative Comparison of Deep Learning Models for AF Detection

Ref	**Architecture**	**Dataset**	**Classes**	**Performance Metrics (AF class)**	**Year**
[14]	34-layer 1D-CNN	Zio Patch Dataset (Proprietary)	14	Sequence F1-score = 0.604, Set F1-score = 0.667	2019
[17]	CNN-LSTM (8CSL)	PhysioNet Challenge 2017	2 (AF / Normal)	F1-score = 89.55%	2020
[16]	1D-CNN + Transformer (ECG DETR)	MIT-BIH AFIB	2 (af / non-af)	Accuracy = 99.23%, F1-score = 99.23%	2022
[15]	Spectrogram + EfficientNet B0	PhysioNet Challenge 2017	2 (AF / Normal)	F1-score = 88.2%, Accuracy = 97.3%	2022
[18]	CWT + Multi-branching ResNet	PhysioNet Challenge 2017 + OUHSC	2 (AF / Normal)	F1-score = 0.8865 (PhysioNet), 0.7396 (OUHSC)	2024
[19]	CNN + Transformer Encoder-Decoder Object Detection (AF-DETR)	CPSC2021, AFDB, LTAFDB, MITDB, NSRDB	3 (AF, Non-AF, Non-obj)	Beat-level F1-score = 96.77% (AFDB), 96.20% (LTAFDB); Segment-level AF F1-score = 97.67% (AFDB), 97.72% (LTAFDB)	2025

2.7 Conclusion

This chapter has provided an overview of ECG analysis with a particular focus on atrial fibrillation detection. Specifically, the creation of the ECG signal electrically was discussed, alongside the basic waveform structures of the ECG and important parameters for diagnosis, in addition to the various issues that prevent the proper automatic analysis of the ECG signal in practice. This was done via extensive reviews of existing literature to demonstrate the transition from machine learning techniques involving feature extraction to deep learning, especially con-

cerning AF detection.

As pointed out in this chapter through identifying the shortcomings of current techniques, in particular, the significant manual pre-processing steps required and the inability to learn long-term dependencies, there has been a need for better methods to be developed. Having discussed the medical background and theory in this chapter, the following chapter will focus on presenting the architecture and implementation of the proposed Deep-ECG system that incorporates both CNNs and Transformer networks for AF classification using single-lead ECG.

CHAPTER 3

DESIGN OF DEEP-ECG SYSTEM

3.1 Introduction

In the current chapter, we introduce the procedure that leads to designing and implementing the Deep-ECG based model aimed at detecting atrial fibrillation. The proposed approach comprises a set of interconnected stages intended to enhance the effectiveness of the whole algorithm. The stages include preparation of input data, preprocessing the ECG signals, extracting features, and model training and evaluation. In summary, this approach provides a full-fledged model able to discriminate between atrial fibrillation and non-AF rhythms.

3.2 Suggested Approach

The Real-Time Atrial Fibrillation Detection System (named Deep-ECG) is developed following a systematic and structured methodology which combines the high quality of three PhysioNet databases and the efficiency of modern deep learning techniques, resulting in a strong and clinically reliable performance. The pipeline has several steps interconnected between each other, guaranteeing that the whole process runs smoothly

from data collection until the predictions made by the system.

In general terms, the process starts with the collection of different ECG signals available in PhysioNet databases CPSC2018, Georgia and ECG-Arrhythmia. These signals compose a large variety of normal sinus rhythm (NSR) and atrial fibrillation (AF) cases. After collecting all the data, it passes through an efficient pipeline of preprocessing. Then, important and clinically related information are extracted from each signal. In this pipeline, we extract information not only related to the morphology of each single beat but also RR-intervals related to the abnormal heart rhythm of AF patients. All the extracted information is fed into our Dual Transformer neural network model, which is capable of combining beat information and RR-intervals in order to better describe each single case. The model is trained and evaluated to test the ability to generalize unseen cases.

In this work, we carefully consider each step of the proposed pipeline to mitigate frequent issues observed in the processing of ECG signals, such as noise, variability among patients, data class unbalance, etc. and exploit the advantages from morphology and rhythm for achieving a high-performance AF predictive system capable of successfully generalizing in unseen data and easily adapted into new clinical environments.

Figure 3.1 shows the flowchart illustrating the process of developing an AF diagnosis system.

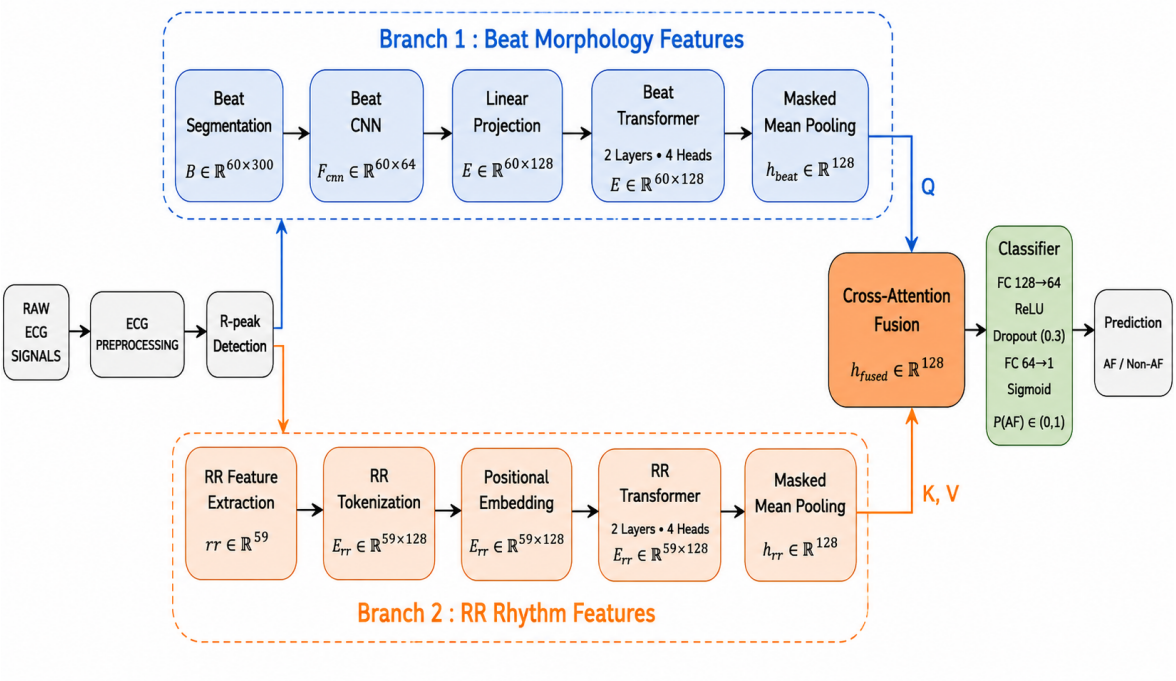


Figure 3.1: Proposed Deep-ECG System for Atrial Fibrillation Detection.

3.3 System Development Phases

We present a multi-stage approach for the proposed system. Each stage has been carefully designed and executed, which improves the quality of the ECG signals we acquire and at the same time produces clinically relevant features. This is a full workflow which begins with data preparation and ECG preprocessing, continues with feature extraction from beat morphology and RR intervals, and concludes with model training and evaluation.

3.3.1 Preprocessing

The Lead II ECG signal is typically contaminated with various forms of noise and artifact which include baseline wander, power-line interference, muscle noise, as well as amplitude spikes; therefore, in order to prepare the raw ECG data for beat segmentation, RR interval determination and subsequently model development, a series of pre-processing steps

were developed that mitigate the primary sources, normalize the cleaned signals.

- **Resampling:** To ensure consistency across the different PhysioNet datasets, all signals were resampled to a uniform target sampling frequency of 500 Hz.
- **Spike Removal:** To suppress extreme amplitude outliers, the signal was clipped at a statistical threshold
- **Bandpass Filtering:** A 4th-order Butterworth bandpass filter was applied to remove baseline wander and high-frequency noise while preserving clinically relevant ECG morphological components (P-wave, QRS complex, and T-wave)
- **Notch Filtering:** An IIR notch filter centred at 50 Hz with a quality factor of $Q = 30$ was applied to suppress power-line interference.
- **Z-score Normalization:** The filtered signal was standardized to have zero mean and unit variance

From each multi-lead recording, only Lead II was selected, as it is known to provide the clearest visualization of atrial activity and is therefore critical for accurate atrial fibrillation detection Figure 3.2 shows the comparison between raw and preprocessed ECG signals for both Normal Sinus Rhythm (NSR) and Atrial Fibrillation (AF) cases.

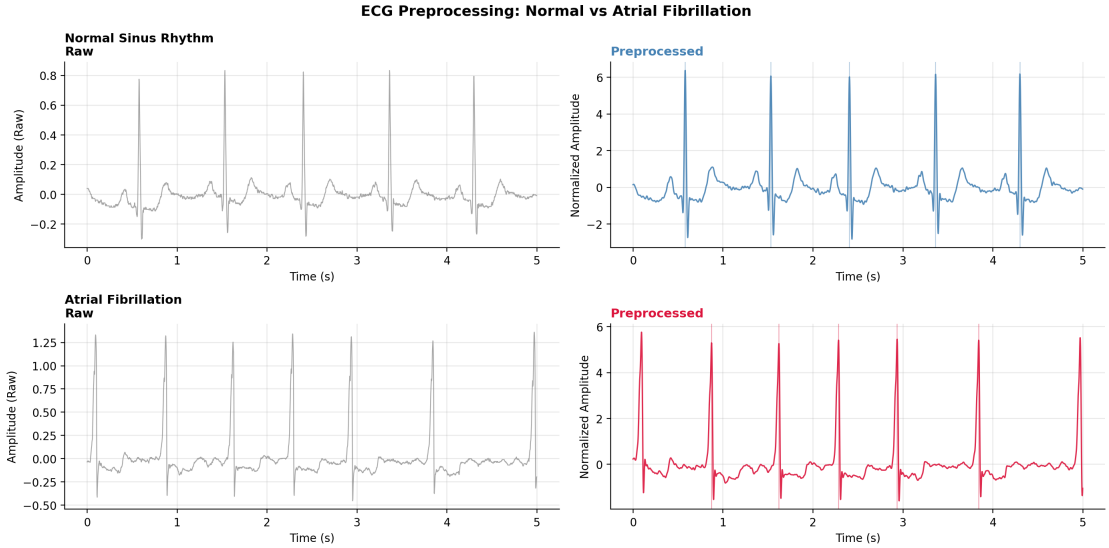


Figure 3.2: ECG preprocessing comparison: raw (left) and preprocessed (right) signals for Normal Sinus Rhythm (top) and Atrial Fibrillation (bottom).

3.3.2 R-Peak Detection

R-peak detection is an intermediate step in order to enable feature extraction for the ECG signal. For the preprocessed signal of Lead II, all R-peaks of the QRS-complex are identified using NeuroKit2's `ecg_peaks` function, which implements a state-of-the-art R-peak detection algorithm optimized for clinical ECG analysis, which is a clinically approved method for an exact R-peak detection. The preprocessed ECG signal of Lead II is sampled at 500 Hz, and the function outputs an array of the sample-indices of the R-peaks $[r_1, r_2, \ldots, r_n]$, with n being the number of the found peaks.

Figure 3.3 illustrates the overall architecture of the R-peak detection step.

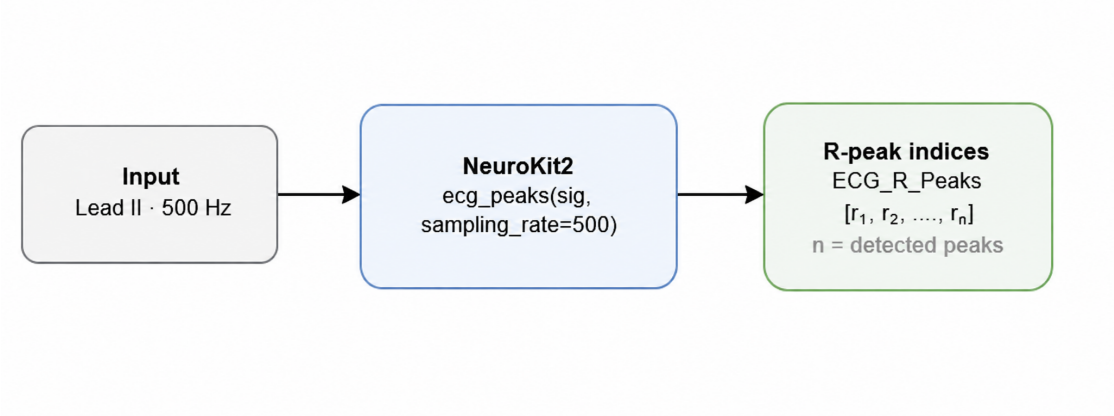


Figure 3.3: R-Peak Detection architecture overview.

3.3.3 Branch 1: Beat Morphology

The Beat Morphology branch encodes fine-grained shape information of single heartbeats into a fixed-size feature vector. This feature vector, called the Beat Morphology (BM) feature, is a compact contextual representation of morphological information of heartbeats in a recording that is used for the classification of Atrial Fibrillation (AF).

Beat Segmentation

This step first extracts individual heartbeats from the preprocessed ECG signal by their R-peaks as references. A fixed-size window of 300 samples (600 ms at 500 Hz) is centred around each detected R-peak r_i. The window is symmetric around r_i, spanning ± 150 samples:

$$\text{Window}_i = [r_i - 150,\ r_i + 150) \tag{3.1}$$

The step then for each such window performs z-score normalization. From a recording up to 60 such individual heartbeats are extracted. The step then performs zero-padding if less than 60 beats of a recording are valid (i.e. were found by the detection step). The output of this step is a so-called beat-matrix of the form:

$$\text{beat}_i = \text{z-norm}(\text{ECG}[r_i - 150 : r_i + 150]) \in \mathbb{R}^{300} \tag{3.2}$$

where r_i is the sample index of the i-th R-peak, and z-norm denotes zero-mean unit-variance normalization.

The extracted beats are stacked into the beat matrix:

$$B = [\text{beat}_1,\ \text{beat}_2,\ \ldots,\ \text{beat}_{60}] \in \mathbb{R}^{60\times 300} \tag{3.3}$$

Beat CNN

For each beat in the input signal a set of morphological features is extracted by a shared one-dimensional convolutional neural network (1D-CNN). The same network weights are used for all 60 beats in a recording and thus all beats are analyzed in the same way, regardless of their position in the signal.

Our proposed architecture is composed of three 1D convolutional blocks followed by an adaptive average pooling, as illustrated in Figure 3.4.

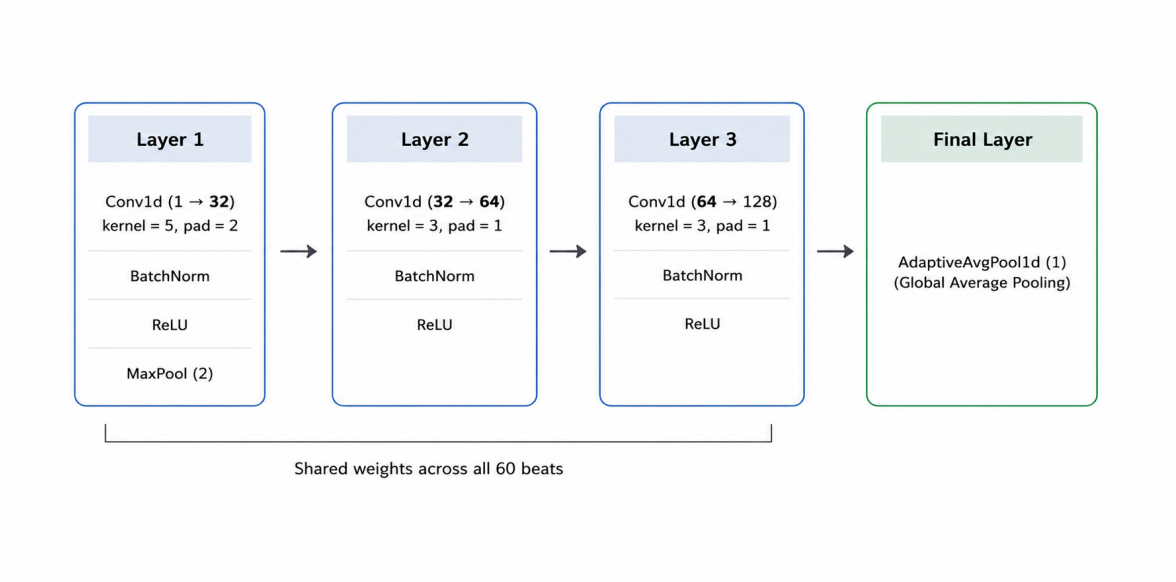


Figure 3.4: Architecture of the Beat CNN.

Block 1 of our model consists of a convolution layer with a kernel size of 5 with padding 2, followed by batch normalization, ReLU activation, and max pooling. The max-pooling layer results in a 50% reduction of the time dimension (from 300 to 150 samples). Blocks 2 and 3 consist of 1D convolutional layers with kernel size 3 amd padding 1 followed by batch normalization + ReLU activation, each of which increases in number of output channels from 32 to 64, and from 64 to 128, respectively. The 128-dimensional feature vector $\mathbf{c}_i$ produced by the final convolutional block (after adaptive average pooling) is then mapped by a fully connected layer to a 64 dimensional embedding:

$$\mathbf{f}_i = W_{\text{cnn}}\mathbf{c}_i + \mathbf{b}_{\text{cnn}} \in \mathbb{R}^{64} \tag{3.4}$$

where $\mathbf{c}_i \in \mathbb{R}^{128}$ represents the convolutional feature vector extracted from beat i.

The final BeatCNN output is therefore:

$$F_{\text{cnn}} = [\mathbf{f}_1,\ \mathbf{f}_2,\ \ldots,\ \mathbf{f}_{60}] \in \mathbb{R}^{60\times 64} \tag{3.5}$$

Linear Projection

The 64-dimensional CNN embeddings are projected to a model dimension of $d_{\text{model}} = 128$ to be suitable as input for the Transformer encoder in the next network layer.

A single shared fully connected linear layer is applied to each beat vector to get the embedding of the beat.

$$\mathbf{e}_i = W_{\text{proj}}\mathbf{f}_i + \mathbf{b}_{\text{proj}} \in \mathbb{R}^{128} \tag{3.6}$$

where:

$$W_{\text{proj}} \in \mathbb{R}^{128\times 64}$$

is the learned projection matrix.

The resulting projected sequence is:

$$E = [\mathbf{e}_1,\ \mathbf{e}_2,\ \ldots,\ \mathbf{e}_{60}] \in \mathbb{R}^{60\times 128} \tag{3.7}$$

Beat Transformer Encoder

Temporal relations between subsequent beats are captured by a Transformer Encoder which utilizes self-attention to model long-range morphological relations in the recording.

Learned positional embeddings are first added to the sequence to preserve the natural order of the ECG beats:

$$E_{\text{in}} = [\mathbf{e}_1 + \mathbf{pos}_1,\ \mathbf{e}_2 + \mathbf{pos}_2,\ \ldots,\ \mathbf{e}_{60} + \mathbf{pos}_{60}] \in \mathbb{R}^{60\times 128} \tag{3.8}$$

The transformer encoder is composed of two stacked layers. Each layer consists of the following sub-layers:

- **Multi-Head Self-Attention:** 4 attention heads, $d_{\text{model}} = 128$, dropout $= 0.2$
- **Add & LayerNorm** residual connection after attention
- **Feed-Forward Network:** Linear$(128 \to 256) \to$ ReLU $\to$ Linear$(256 \to 128)$
- **Add & LayerNorm** residual connection after FFN

During self-attention of the encoder layers, the padding mask of the beats is applied, so that no information is extracted from the padding of the beats.

The contextual beat sequence is obtained as:

$$E' = \text{BeatTransformerEncoder}(E_{\text{in}}) \in \mathbb{R}^{60\times 128} \tag{3.9}$$

Beat Representation (Masked Mean Pooling)

The contextual beat vectors from the last layer of the encoder are aggregated into a fixed-size vector that encodes the overall morphological content of a recording.

In contrast to mean pooling, we employ masked mean pooling to ignore the beats that have been padded for a recording in the beat representation.

The final beat representation is computed as:

$$\mathbf{h}_{\text{beat}} = \frac{\sum_{i=1}^{60} m_i \mathbf{e}'_i}{\sum_{i=1}^{60} m_i} \in \mathbb{R}^{128} \tag{3.10}$$

where $m_i \in \{0, 1\}$ is the validity mask indicating whether beat i is real or padded.

The resulting vector $\mathbf{h}_{\text{beat}} \in \mathbb{R}^{128}$ serves as the query representation for the cross-attention fusion module.

Figure 3.5 illustrates the complete pipeline of Branch 1, from the raw beat matrix through BeatCNN, linear projection, positional encoding, Transformer encoder, and masked mean pooling to produce the final beat representation.

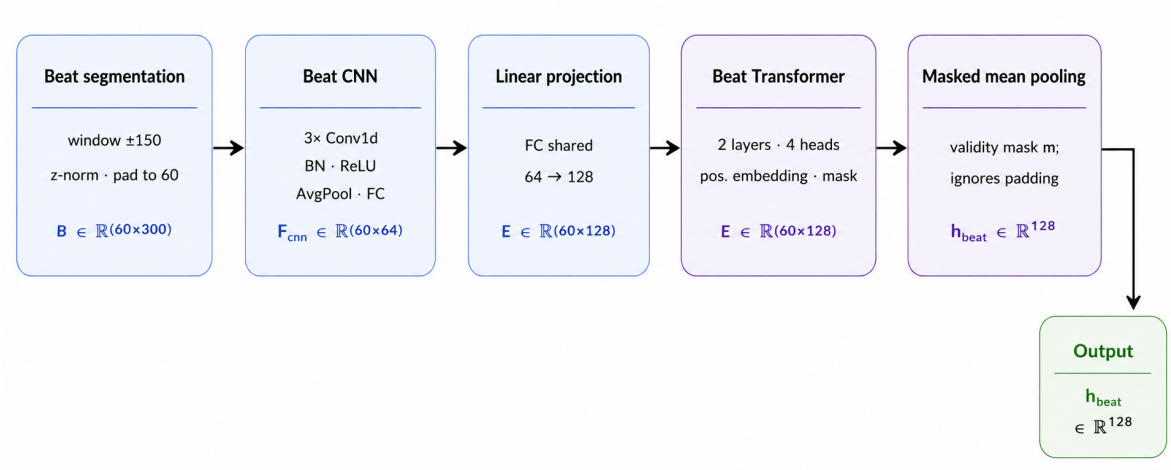


Figure 3.5: Branch 1 (Beat Morphology)architecture

3.3.4 Branch 2: RR Rhythm Features

The RR Rhythm branch captures the temporal dynamics of heart rate variability (HRV),especially the irregular ventricular response characteristic of Atrial Fibrillation. The RR interval sequence is tokenized and processed by a Transformer encoder to produce a 128-dimensional rhythm representation $\mathbf{h}_{rr} \in \mathbb{R}^{128}$.

RR Feature Extraction

The RR intervals are calculated as the difference in time between each pair of successive R-peaks. Intervals outside of the physiological range of 0.3 s to 2.0 s are removed from the data. The valid RR intervals are then padded with zeros to create a fixed-length sequence of 59 intervals, with shorter sequences zero-padded and longer sequences truncated:

$$RR_i = \frac{r_{i+1} - r_i}{f_s}, \qquad 0.3 \leq RR_i \leq 2.0 \text{ s} \tag{3.11}$$

where r_i are consecutive R-peak sample indices and $f_s = 500$ Hz. The resulting RR sequence is:

$$\mathbf{rr} = [RR_1, \ldots, RR_{59}] \in \mathbb{R}^{59} \tag{3.12}$$

RR Tokenization

Each scalar RR interval is mapped onto the transformer embedding space through a single shared linear layer. The conversion of the raw temporal measurements into dense vectors that can be processed using self-attention occurs in this tokenization step:

$$\mathbf{tok}_i = W_{\text{rr_tok}} \cdot RR_i + \mathbf{b}_{\text{rr_tok}} \in \mathbb{R}^{128} \tag{3.13}$$

where $W_{\text{rr_tok}} \in \mathbb{R}^{128\times 1}$ is the learned projection matrix. The resulting token sequence is:

$$E_{\text{rr}} = [\mathbf{tok}_1,\ \mathbf{tok}_2,\ \ldots,\ \mathbf{tok}_{59}] \in \mathbb{R}^{59\times 128} \tag{3.14}$$

Positional Embedding & Masking

Learned positional embeddings are added to each RR token to preserve temporal order. A boolean padding mask is also computed to ensure zero-padded positions do not participate in self-attention:

$$\mathbf{e}_{\text{rr},i} = \mathbf{tok}_i + \mathbf{pos}_i \in \mathbb{R}^{128} \tag{3.15}$$

$$m_i = \begin{cases} \text{True} & \text{if } RR_i \leq 0 \quad \text{(padded)} \\ \text{False} & \text{otherwise} \end{cases} \tag{3.16}$$

The enriched token sequence and its corresponding mask are:

$$E_{\text{rr}} = [\mathbf{e}_{\text{rr},1},\ \mathbf{e}_{\text{rr},2},\ \ldots,\ \mathbf{e}_{\text{rr},59}] \in \mathbb{R}^{59\times 128} \tag{3.17}$$

$$\mathbf{m} = [m_1,\ m_2,\ \ldots,\ m_{59}] \in \{\text{True},\ \text{False}\}^{59} \tag{3.18}$$

RR Transformer Encoder & Masked Mean Pooling

We experiment with a 2-layer Transformer encoder of hidden size 128 with 4 attention heads of size 32 and dropout = 0.2. This Transformer is applied to the 59-token RR sequence after the padding mask has been applied to it.

$$H_{\text{rr}} = \text{RRTransformerEncoder}(E_{\text{rr}},\ \mathbf{m}) \in \mathbb{R}^{59\times 128} \tag{3.19}$$

Masked mean pooling then averages only over valid (non-padded) positions to produce the sequence representation:

$$\mathbf{h}_{\text{seq}} = \frac{\sum_{i=1}^{59} \mathbb{1}_{[\neg m_i]} \cdot H_{\text{rr}}^{(i)}}{\sum_{i=1}^{59} \mathbb{1}_{[\neg m_i]}} \in \mathbb{R}^{128} \tag{3.20}$$

where $\mathbb{1}_{[\neg m_i]}$ is the indicator function for valid positions ($m_i = \text{False}$).

RR Representation

The sequence representation obtained from masked mean pooling serves directly as the final rhythm representation:

$$\mathbf{h}_{rr} = \mathbf{h}_{seq} \in \mathbb{R}^{128} \tag{3.21}$$

The resulting vector $\mathbf{h}_{rr} \in \mathbb{R}^{128}$ serves as the Key and Value in the cross-attention fusion module.

Figure 3.6 illustrates the complete pipeline of Branch 2, from the raw RR feature vector through tokenization, positional encoding, Transformer encoder, masked mean pooling, and final projection to produce $\mathbf{h}_{rr}$.

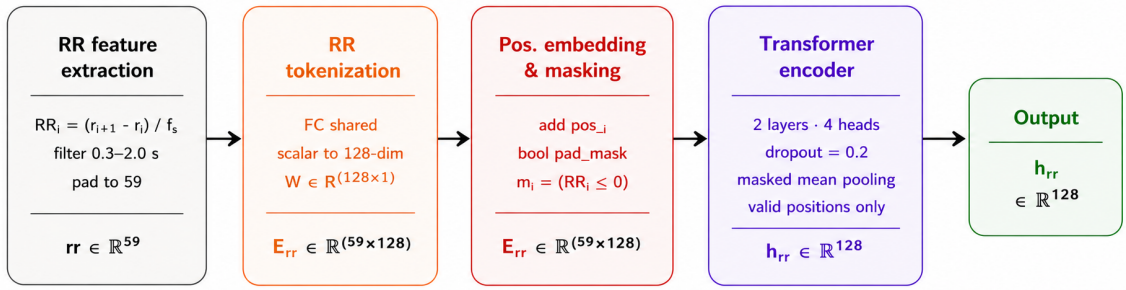


Figure 3.6: Branch 2 (RR Rhythm) architecture

3.3.5 Cross-Attention Fusion

The morphological information of Branch 1 and the rhythmic information of Branch 2 are combined in the final layer by means of a cross-attention mechanism. Instead of concatenating both representations in a naive way, we first select the most relevant parts of the rhythm representation for every single recording based on the morphological context,

i.e. the beat representation $\mathbf{h}_{\text{beat}}$ serves as the Query whereas $\mathbf{h}_{\text{rr}}$ is used for both the Key and the Value representation. After reshaping the two vectors to single-token sequences, they are passed through the attention mechanism:

$$Q = \mathbf{h}_{\text{beat}}^{\top}, \quad K = V = \mathbf{h}_{\text{rr}}^{\top} \in \mathbb{R}^{1\times 128} \tag{3.22}$$

$$\mathbf{x} = \text{LayerNorm}(Q + \text{Dropout}(\text{MHA}(Q, K, V))) \tag{3.23}$$

$$\mathbf{h}_{\text{fused}} = \text{LayerNorm}(\mathbf{x} + \text{Dropout}(\text{FFN}(\mathbf{x}))) \in \mathbb{R}^{128} \tag{3.24}$$

where $\text{FFN} = \text{Linear}(128\to128) \to \text{ReLU} \to \text{Dropout}(0.2) \to \text{Linear}(128\to128)$. The residual connection from the output is Q (i.e. $\mathbf{h}_{\text{beat}}$) to ensure that the basic morphological information about the input ECG has been retained, even when the attention weights from the rhythmic features are small.

3.3.6 Classifier

The fused 128-dimensional representation is then mapped by a multilayer perceptron to a single real value, the probability that a sample belongs to the class of Atrial Fibrillation, after applying dropout regularization:

$$P(\text{AF}) = \sigma(W_2\,\text{Dropout}(0.3)(\text{ReLU}(W_1\,\mathbf{h}_{\text{fused}} + \mathbf{b}_1)) + \mathbf{b}_2) \in [0, 1] \tag{3.25}$$

where $W_1 \in \mathbb{R}^{64\times128}$, $W_2 \in \mathbb{R}^{1\times64}$, and $\sigma(\cdot)$ denotes the sigmoid function. The binary decision rule is:

$$\hat{y} = \begin{cases} 1 \text{ (AF)} & \text{if } P(\text{AF}) \geq 0.5 \\ 0 \text{ (Non-AF)} & \text{if } P(\text{AF}) < 0.5 \end{cases} \tag{3.26}$$

Figure 3.7 illustrates the cross-attention fusion module and the classifier pipeline.

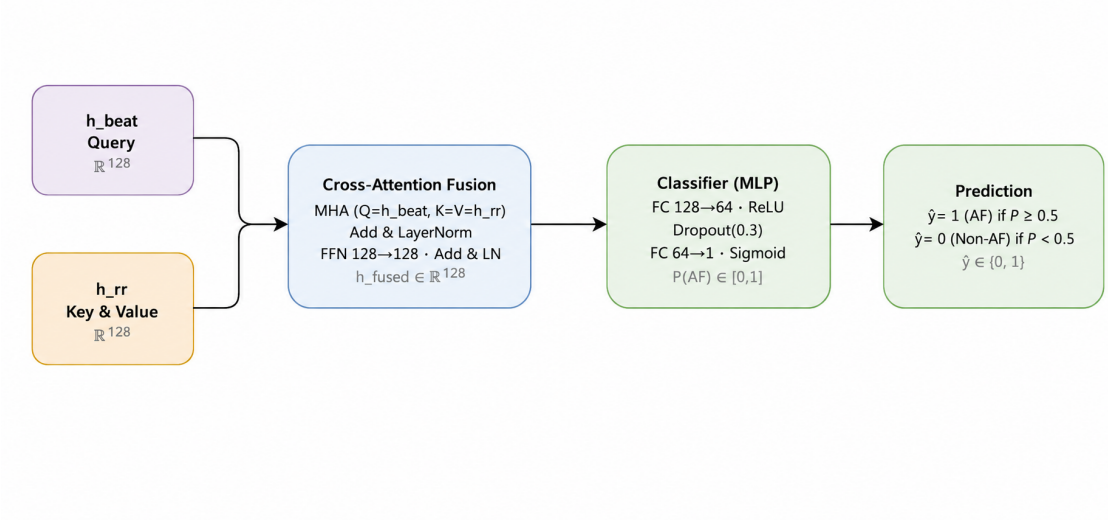


Figure 3.7: Cross-Attention Fusion and Classifier architecture

3.4 Conclusion

This chapter presented a conceptual methodology for developing the deep learning-assisted detection of atrial fibrillation. The chapter showcased the structure of the proposed Deep-ECG (a dual-branch Transformer deep learning model) which aimed to combine the complementary nature of two streams (beat-level morphological and RR interval rhythm) information for AF detection.

The PhysioNet databases (CPSC2018, Georgia, and ECG-Arrhythmia) provided the ECG dataset used in this work. Each database was preprocessed prior to model development. Spike removal, bandpass filtering, notch filtering, normalization, and resampling to 500 Hz are part of the preprocessing pipeline.

Beat-level morphological features were extracted from the preprocessed ECG signal, and RR interval features were derived to capture the rhythm dynamics. In an end-to-end manner, the final decision is made by combining the two branches via cross-attention. The model was implemented in PyTorch and trained using mixed-precision (AMP) for efficient GPU utilisation.

The technical foundations established in this chapter provide a solid basis for the experimental evaluation presented in the following chapter.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter describes the experimental evaluation and validation of the proposed Deep-ECG detailed in the previous chapter. To demonstrate the efficacy of the hybrid dual-branch transformer architecture in the detection of atrial fibrillation, various experiments were conducted.

The work begins with a brief overview of the development environment, hardware and software dependencies, model implementation details and training hyperparameters. Finally, the system is evaluated and compared with state-of-the-art methods using a set of standard clinical performance metrics.Following the quantitative evaluation, the chapter delves into the explainability layers that provide clinical transparency, and concludes with the practical deployment of the framework through a dedicated medical web application.

4.2 Development Environment

4.2.1 Hardware Specifications

Google Colab is a cloud-based platform that offers Jupyter notebooks by Google where you can execute the code through your web browser and have access to high-performance computing resources. Google Colab was used to execute all the preprocessing and model evaluations for our experiments. The experiments were conducted using the free version of Google Colab, with GPU hardware acceleration enabled using an NVIDIA Tesla T4 GPU, allowing us to perform the high-performance tensor calculations necessary for deep learning models and physiological signal processing for the ECG data.

4.2.2 Python Modules

Python (version 3.12.13) For the experimental workflows of the system, the primary programming language used was Python. Python is very efficient for scientific computing. There are many libraries that can be used for data processing, machine learning and signal analysis, etc[20].

The key capabilities of the system were provided by the following specialized libraries:

- **PyTorch (Version 2.10.0+cu128)**: The high-performance tensor library is designed to perform efficiently in CPU as well as GPU environments. In the current experiment, PyTorch was employed for the design and implementation of the proposed hybrid dual-branch deep learning architecture[21].
- **SciPy (Version 1.16.3)**: SciPy is a collection of mathematical algorithms for scientific computing built on top of NumPy, and designed to encompass advanced mathematics and signal processing.

This was used for simulating only digital signal processing techniques here; specifically, Butterworth filter in bandpass and notch configuration to eliminate ECG noise[22].

- **NeuroKit2 (Version 0.2.13)**: In this work, we have used NeuroKit2 for ECG preprocessing, namely for R-peak detection and for segmentation of individual heartbeats (cardiac cycles) in order to analyze them in detail[23].
- **WFDB (Version 4.3.1)**: WFDB is a specialized software library for reading and writing signal data and associated annotations as well as for processing such data. In our work WFDB was utilized to read ECG data from PhysioNet databases as well as corresponding annotations[24].

4.3 Datasets

The data used in this work is a combination of three publicly available ECG databases. For each database, the labels for the recordings were extracted from the `#Dx:` field in the `.hea` header file for that recording. In each label was then matched against two SNOMED-CT codes: `164889003` for Atrial Fibrillation and `426783006` for Normal Sinus Rhythm. Only recordings labeled as Normal Sinus Rhythm, carrying only the Normal code, were used as the negative examples for the data. Other recordings, which had other diagnoses in addition to the Normal code, were discarded. In addition to label filtering, each recording was also rejected if it did not have a Lead II channel, had fewer than 100 samples, had a near constant signal (std below 10^{-6}), or had fewer than 5 R-peaks that were detected in the preprocessed signal.

Dataset 1: CPSC2018

The China Physiological Signal Challenge 2018 (CPSC2018) dataset contains 6,877 12-lead ECG recordings,lasting from 6 seconds to 60 seconds. Each recording was sampled at 500 Hz, collected in Chinese hospitals. In total, 2,139 recordings (918 Normal and 1221 AF) remain after applying the filtering and quality control criteria. The local prevalence of AF in this subset is 56.8% [25].

Dataset 2: Georgia (PhysioNet Challenge 2020)

The Georgia subset of the PhysioNet/Computing in Cardiology Challenge 2020 contains 10,344 recordings Of these recordings, 2,321 were accepted for inclusion in this study,of 10 second length with a sampling frequency of 500 Hz. There were 1,752 recordings labeled as Normal and 569 recordings labeled as AF, for an average prevalence of AF of 24.5%. This set of recordings had the highest rejection rate of the three datasets. This is largely due to the large number of multi-label recordings in this database [25].

Dataset 3: ECG-Arrhythmia (PhysioNet)

The ECG-Arrhythmia database, the largest of the three databases used in this work, contains 45,152 recordings with a sampling frequency of 500 Hz and taken over a period of 10 seconds. After applying the criteria to these recordings 7,688 were retained, i.e. 5,908 Normal and 1,780 recordings with AF, resulting in an AF prevalence of 23.2%. The rejection rate for this database is approximately 83% and it is almost entirely due to the large amount of multi-label recordings contained in this database [26].

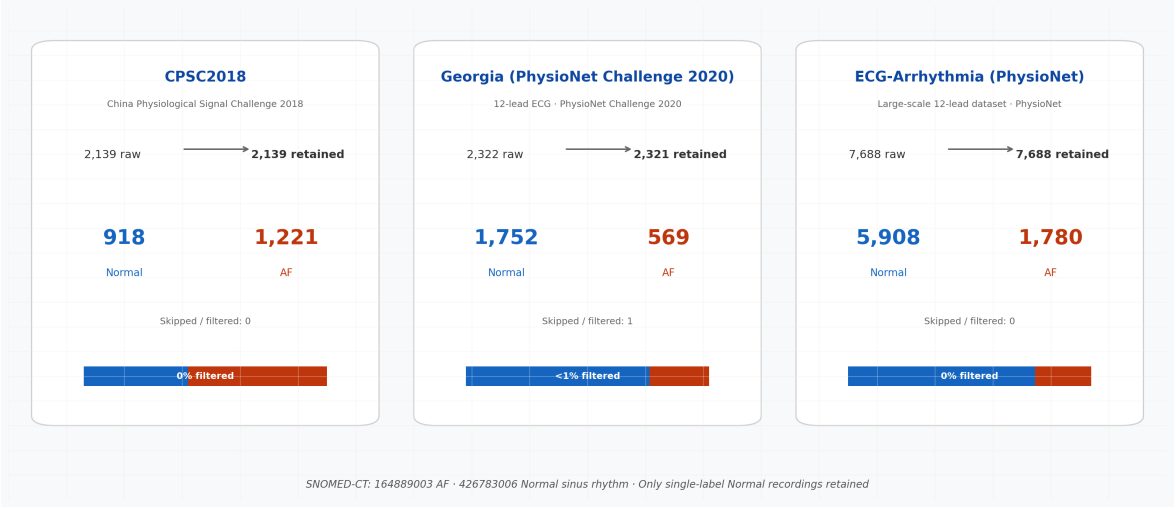


Figure 4.1: Summary of the three ECG data sources after filtering.

Combined Dataset and Class Balance

The three databases contained a total of 12,148 recordings. Statistics of the pooled dataset are shown in Figure 4.2. The pooled dataset comprises 8,578 Normal recordings (70.6%) and 3,570 AF recordings (29.4%). The class imbalance ratio is approximately 2.4:1 in favour of the Normal class, indicating a substantial over-representation of Normal recordings in the pooled dataset. To address this class imbalance during model training, we employ Binary Cross-Entropy with logits and positive class weighting:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[w^{+} y_i \log \sigma(\hat{z}_i) + (1 - y_i) \log\left(1 - \sigma(\hat{z}_i)\right)\right] \tag{4.1}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function, $\hat{z} * i$ is the raw logit output of the classifier for recording i, $y_i \in 0, 1$ is the ground-truth label (1 for AF and 0 for Normal), and $w^{+} = \frac{n*\text{neg}}{n_{\text{pos}}}$ is the positive class weight computed from the training set class frequencies.

In practice, the loss is implemented using `BCEWithLogitsLoss`, which internally applies the sigmoid activation and combines it with binary

cross-entropy in a numerically stable formulation. The positive class weight ensures that AF recordings contribute proportionally more to the optimisation process, thereby reducing the bias toward the majority Normal class.

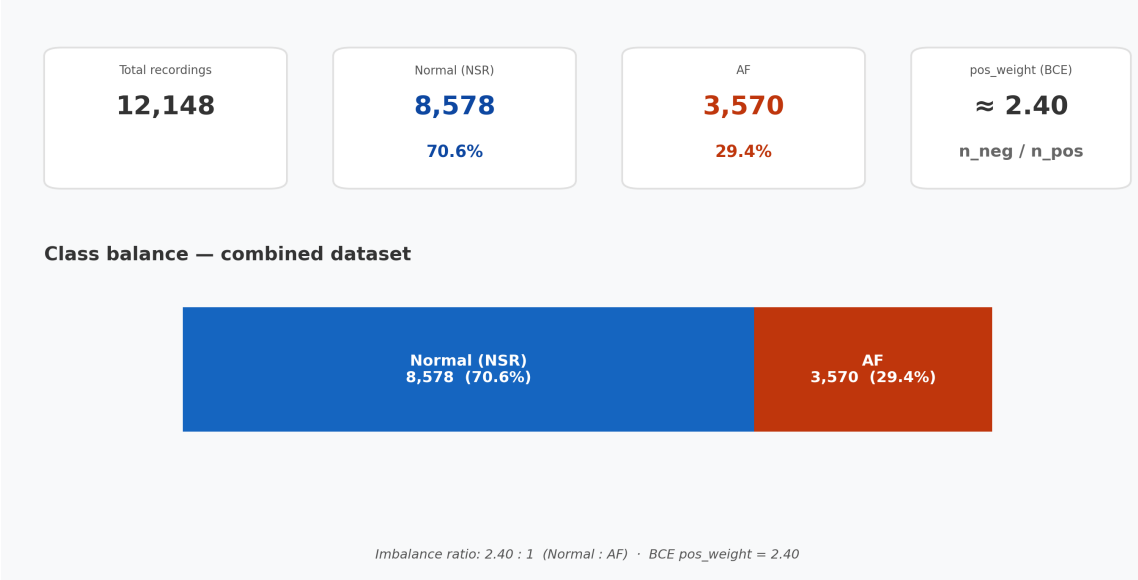


Figure 4.2: Combined dataset statistics and class balance.

Train / Validation / Test Split

The dataset was then split in a stratified manner to 70% training, 15% validation and 15% test using a fixed random seed.This split was chosen to provide sufficient data for model training while reserving independent validation and test sets for unbiased hyperparameter tuning and performance evaluation. As can be seen in the figure 4.3, the percentage of AF recordings were also preserved in the same manner for all three splits.

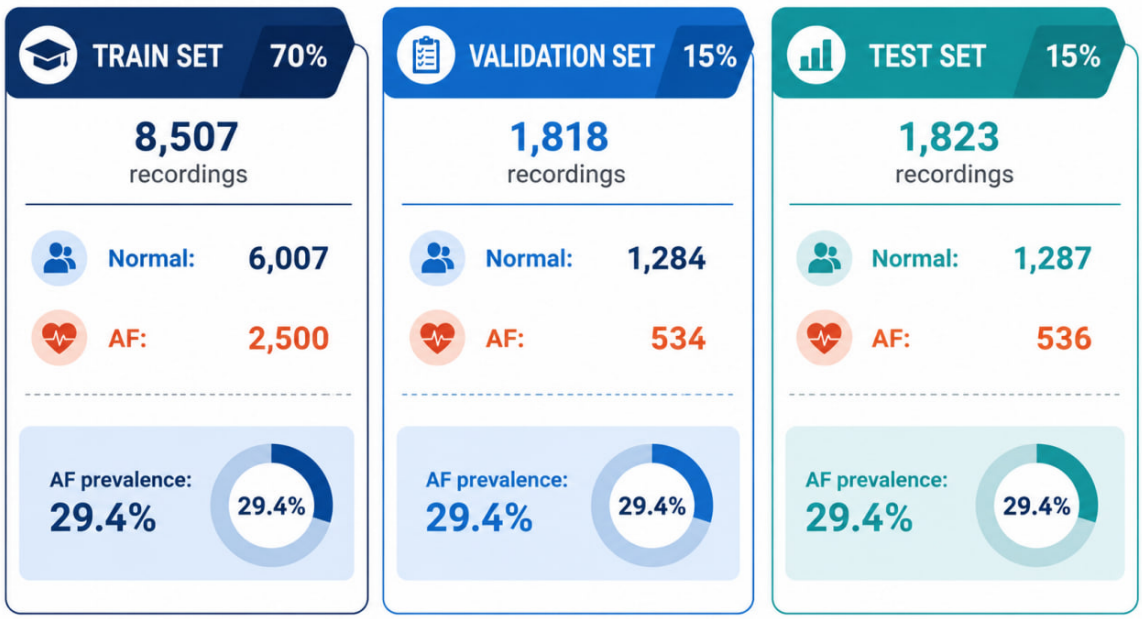


Figure 4.3: Stratified train/validation/test split.

4.4 Implementation and Results

4.4.1 Experimental Setup

All experiments conducted in this work were done under identical training conditions to allow for a fair comparison and reproducibility of results; a summary of the set-up can be seen in the table below 4.1.

Table 4.1: Summary of shared training hyperparameters for all proposed models.

Hyperparameter	**Value**
Optimizer	Adam
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Loss function	BCEWithLogitsLoss
Positive class weight (w^+)	≈ 2.40 (6,007 / 2,500)
Batch size	32
Gradient accumulation steps	4
Effective batch size	128
Gradient clipping (max norm)	1.0
LR scheduler	ReduceLROnPlateau
LR reduction factor	0.5
LR scheduler patience	3 epochs
Maximum epochs	60
Early stopping patience	10 epochs
Monitored metric (checkpoint)	Validation loss
Mixed precision (AMP)	Enabled

4.4.2 Evaluation Metrics

The performance of all models was assessed by measuring the accuracy of the above metrics on a sample set out of a total population of 1823 samples. Due to the class imbalance in the sample set, with 70.6% of the samples being classified as Normal and 29.4% of the samples being classified as AF, only one metric (accuracy) will be insufficient to evaluate model performance. Thus, the subsequent metrics were used to provide a thorough assessment of each model's performance,

Let TP, TN, FP, and FN denote the number of True Positives, True Negatives, False Positives, and False Negatives, respectively, Where the positive class represents AF.

Accuracy measures the overall proportion of correctly classified recordings:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

Precision measures the proportion of recordings predicted as AF that are truly AF:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.3}$$

Recall (Sensitivity) measures the proportion of true AF recordings that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.4}$$

Specificity measures the proportion of Normal recordings that are correctly identified as Normal, which is particularly important in a medical diagnosis context where minimizing false alarms is critical:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.5}$$

F1-Score is the harmonic mean of Precision and Recall, providing a single balanced measure that accounts for both false positives and false negatives:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.6}$$

AUC-ROC (Area Under the Receiver Operating Characteristic Curve) measures the model's ability to discriminate between the AF and Normal classes across all classification thresholds. A value of 1.0 indicates perfect discrimination, while a value of 0.5 corresponds to random chance.

4.4.3 Performance of Deep-ECG system

The Dual-Branch Cross-Attention Model (Deep-ECG) comprises 701,441 parameters and was trained for 53 epochs with early stopping, achieving a minimum validation loss of 0.0843. Figure 4.4 illustrates the training dynamics: the validation AUC stabilised above 0.99 within the first few epochs, and the validation accuracy converged to approximately 98.5%, indicating strong generalisation without overfitting.

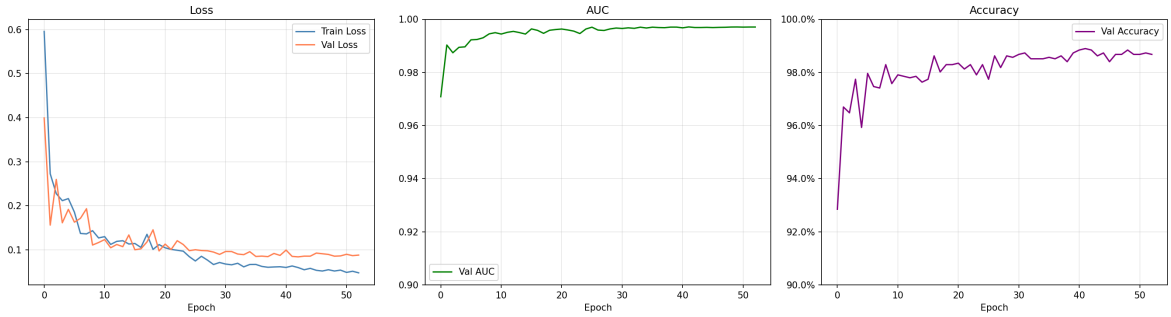


Figure 4.4: Training and validation curves for the Deep-ECG Model: loss (left), AUC (centre), and accuracy (right).

On the held-out test set of 1,823 recordings (1,287 Normal and 536 AF), the model achieved an AUC of 0.9950, an accuracy of 99.18%, and an F1-score of 0.9859.Table 4.2 summarises the complete set of performance metrics.

Table 4.2: Detailed performance metrics of Deep-ECG on the test set.

Model	**AUC-ROC**	**Accuracy**	**Precision (AF)**	**Recall (AF)**	**Specificity**	**F1-Score (AF)**	**FPR**	**FNR**
Deep-ECG	0.9950	99.18%	99.06%	98.13%	99.61%	0.9859	0.39%	1.87%

The confusion matrix and ROC curve are presented in Figure 4.5.

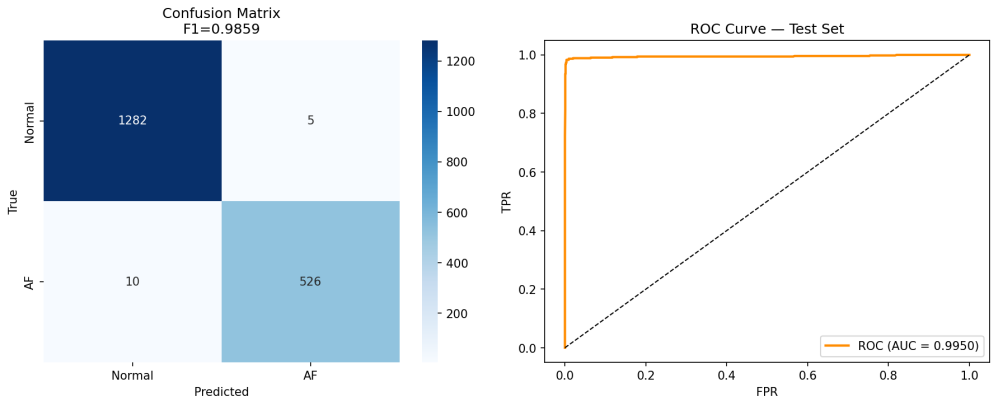


Figure 4.5: Confusion matrix (left) and ROC curve (right) for DEEP-ECG

Error Analysis. The model misclassified only 15 out of 1,823 test recordings (0.82% error rate). The 5 false positives and 10 false negatives warrant closer inspection.

False Positives (5 cases). These recordings exhibited rhythm irregularities or motion artefacts that mimicked AF patterns. Visual inspection suggested transient rhythm variations and signal disturbances resembling AF-like behaviour, making them difficult to distinguish from true AF cases.

False Negatives (10 cases). The majority presented with relatively organised atrial activity and mildly irregular RR intervals, placing them close to the AF/Normal decision boundary. Some recordings showed low-amplitude fibrillatory waves with minimal rhythm irregularity, reducing discriminative signal for both branches.

4.4.4 Comparative Analysis of Implemented Architectures

Table 4.3 summarises the test set performance of all three implemented models under identical experimental conditions.

Table 4.3: Comprehensive performance comparison of the implemented models on the test set.

Model	**Params**	**Acc.**	**Prec.**	**Rec.**	**Spec.**	**F1**	**AUC**
CNN-Transformer Baseline	292K	95.06%	88.85%	95.15%	95.03%	91.89%	98.47%
Dual-Branch Gated Fusion	705K	98.68%	97.23%	98.32%	98.83%	97.77%	99.61%
Deep-ECG	**701K**	**99.18%**	**99.06%**	**98.13%**	**99.61%**	**98.59%**	**99.50%**

A large difference exists between the single-branch Baseline model and the dual-branch architectures in terms of classification accuracy. The Baseline achieved approximately 95% accuracy because it processes the ECG signal as a whole rather than explicitly separating local beat morphology from global rhythm characteristics. This makes it more difficult to effectively model the relationship between these complementary sources of information. Conversely, the dual-branch approach explicitly captures both morphology and rhythm information through dedicated processing branches, which aligns with the clinical assessment of atrial fibrillation based on morphological abnormalities and rhythm irregularities.

The Cross-Attention architecture performs a little better than Gated Fusion with respect to overall performance (99.18% vs. 98.68% accuracy) when comparing the two dual-branch architectures. The difference between the two systems is the way that they fuse the two representations. Gated Fusion uses a static method of using element-wise weightings to combine the two representations, while Cross-Attention allows for dynamic interactions between morphology and rhythm information. Within this system, the beat representation can dynamically focus on

the best possible set of features for the rhythm of an individual recording. As such, this allows for greater flexibility with regard to borderline instances where rhythm information by itself is not enough to accurately classify the recording.

The number of false positives drops steadily: 64 with the Baseline, 15 with Gated Fusion, and only 5 with Cross-Attention. This means Cross-Attention is the safest choice for screening, as it rarely sends healthy patients for unnecessary checks.

Table 4.4: Error distribution comparison across implemented models.

Model	**FP**	**FN**	**Total Errors**	**Error Rate**
CNN-Transformer Baseline	64	26	90	4.94%
Dual-Branch Gated Fusion	15	9	24	1.32%
Deep-ECG	5	10	15	0.82%

The two models have similar parameter counts (701K vs. 705K) and comparable computational requirements. Although the Gated Fusion model achieves a slightly higher AUC (99.61% vs. 99.50%), the Cross-Attention model provides superior overall classification performance, achieving higher accuracy, precision, specificity, and F1-score while maintaining a comparable computational footprint. Therefore, the proposed Deep-ECG model offers the most favourable balance between diagnostic performance and inference efficiency, making it a strong candidate for real-world AF screening applications.

4.4.5 Comparison of Proposed Model with Related Works

Table 4.5 compares the proposed Deep-ECG model with several recent deep learning approaches for atrial fibrillation detection. Several important observations can be drawn from this comparison.

Table 4.5: Quantitative Comparison of Deep Learning Models for AF Detection

Ref	**Architecture**	**Datasets**		**Classes**	**Performance Metrics (AF class)**
		Train	**Test**		
[14]	34-layer 1D-CNN	Zio Patch (90%)	Zio Patch (Held-out)	14	Seq F1 = 0.604, Set F1 = 0.667
[15]	Spectrogram + EfficientNet-B0	PhysioNet 2017 (90%)	PhysioNet 2017 (10%)	2 (AF/ Normal)	F1 = 88.2%, Acc = 97.3%
[18]	CWT + Multi-Branch ResNet	PhysioNet 2017 (80%), OUHSC (80%)	PhysioNet 2017 (20%), OUHSC (20%)	2 (AF/ Normal)	F1 = 0.8865 (PhysioNet), 0.7396 (OUHSC)
[17]	CNN-LSTM (8CSL)	PhysioNet 2017 (70%)	PhysioNet 2017 (30%)	2 (AF/ Normal)	F1 = 89.55%
[19]	AF-DETR (CNN + Transformer Encoder-Decoder)	CPSC 2021	AFDB, LTAFDB, MITDB, NSRDB	3 (AF, Non-AF, Non-Object)	Beat F1 = 96.77% (AFDB), 96.20% (LTAFDB); Segment F1 = 97.67% (AFDB), 97.72% (LTAFDB)
Ours	**Deep-ECG**	**CPSC2018, Georgia, ECG-Arrhythmia**	**Held-out test split from the combined dataset**	**2 (AF/ Normal)**	**F1 = 98.59%, Acc = 99.18%**
[16]	1D-CNN + Transformer	MIT-BIH AFIB	MIT-BIH AFIB (10-fold CV)	2 (AF/Non-AF)	Acc = 99.23%, F1 = 99.23%

Data Accessibility and Diversity. A number of past studies used either private or one public database (proprietary databases) causing poor generalizability and reproducibility. Unlike many of these studies, the proposed model was trained and evaluated using three independent publicly available datasets (CPSC2018, Georgia, and ECG-Arrhythmia), thereby providing greater representation of different patient populations and techniques, and allowing for complete reproducibility of the results with respect to all of the data being studied.

Feature Representation. Currently, there are several methods that utilize either spectrogram-based or wavelet-based representations, which typically require a further transformation to the signal prior to training the model. Though these methods may be useful, they can result in the loss of detailed time-related information for the signal and increased amount of preprocessing required before the model is trained. In contrast, the current model works directly with beat morphology and RR interval sequences, thus retaining clinically significant information about both the cardiac morphology and rhythm dynamics.

Fusion Strategy. An important element of the proposed architecture that distinguishes it from current multi-branch architectures is the unique fusion mechanism applied. Current multi-branch architectures utilize standard fusion mechanisms, such as concatenation and element-wise weighting. The proposed implementation utilizes a cross-attention mechanism that allows for dynamic interactions between morphology and rhythm features; therefore, the proposed system can give attention to the most relevant rhythm pattern for a recording, leading to the high specificity (99.61%) and very low false-positive rate (0.39%) of the proposed system.

Clinical Suitability. In addition to performance, real-world systems that screen for AF should minimize false alerts while retaining effective sensitivity to detecting AF. The proposed Deep-ECG demonstrates high classification performance with an F1-score of 98.59%, accuracy of 99.18%, and a false-positive rate of only 0.39%. The combination of low inference latency and high sensitivity makes this model an attractive candidate for potential use in large-scale clinical screening and real-time monitoring applications.

4.4.6 Explainability

An AI diagnostic tool can be integrated into healthcare if it provides transparency and interpretability. Thus, predicting a given outcome only satisfies the physician's requirements when the rationale behind that outcome can be understood. To meet this need, the Deep-ECG system contains an Explainability Layer that generates visualisations and quantitative information associated with each prediction because both data sets are automatically provided to the model used to generate the predictions. The last convolution layer of the BeatCNN Branch (cnn[7], the third Conv1d block) uses the Gradient-weighted Class Activation Mapping (Grad-CAM) method to generate a per beat window saliency map based on the 300 samples of the specific ECG beat signal window. The result from applying Grad-CAM will have sample-wise saliency information included in each 300-sample beat window signal generated by BeatCNN. The Grad-CAM saliency signal associated with each beat window is interpolated to the original size of 300 samples, and the data is normalised between 0.0 and 1.0. The ECG signal will contain a colour overlaid on the P-wave region that will indicate whether a P-wave is present or absent (green = present, red = absent). The absence of P-waves in an ECG signal is the main clinical marker of A-fib; therefore,

the colouring in the P-wave region provides for easy visual confirmation of a P-wave's presence or absence in the ECG signal. As illustrated in Figure 4.6, beats with detected P-waves are highlighted in green, while those without are marked in red, with the most salient beat selected for detailed analysis framed in orange.

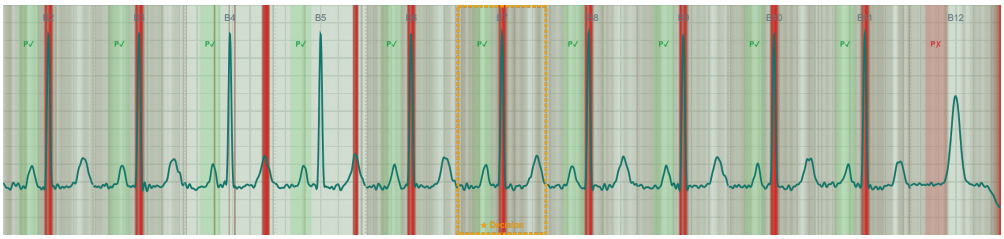


Figure 4.6: ECG visualisation strip showing 11 beats with P-wave colour annotations (green = present, red = absent) and the selected best beat highlighted by an orange dashed frame.

Beyond morphology, the attention weights of the RR transformer encoder indicate which inter-beat intervals contributed most to the detected rhythm irregularity. The interval with the highest attention score can then be readily available to the analysis panel, allowing the physician to validate rhythm evidence based on the information displayed.

Together, the Grad-CAM visualisation, Transformer attention weights, P-wave detection counts, and HRV statistics form a complementary set of explanations that cover both the beat-level morphology and the overall rhythm. This multi-layered approach ensures that the model's output can be cross-checked from several clinical angles, making the system suitable for use as a decision-support tool in practice.

4.4.7 Web Application – Deep-ECG Platform

A web-based interface that facilitates access to a trained model for the analysis of medical data has been developed by SoftNest Technologies. The web application allows for a full workflow of the analysis process: upload one or multiple ECG files, run the model on the uploaded ECG

files, view results after the model has completed and generate a report. The entire workflow can be accomplished in one web browser session and users are not required to have any experience with computer technology to be able to use the system.

System Architecture

The architecture of the system is built upon a client-server model, with a REST API backend (`test.py`) implemented in Flask and run locally on port 5002. Upon receipt of a raw CSV ECG file from the user on the front-end of the system, the backend executes the complete Deep-ECG inference pipeline on the uploaded ECG file(s) and returns the output as a JSON object to the front-end of the system. The JSON object from the server contains the following data: AF probability, Grad-CAM vectors, RR Attention weights, P-wave flags and HRV statistics. The front-end of the system consists of a set of lightweight HTML pages (login, dashboard, analysis, archive, report) sharing a common stylesheet and a single JavaScript file (`app.js`), which requests data from the backend API via local HTTP calls, displays results in the form of ECG visualisations on the webpage with the HTML5 Canvas API, and saves the patient record using the `localStorage` object of the user's web browser.

Dashboard Overview

Upon logging into the platform, the physician is directed to the dashboard. As illustrated in Figure 4.7, this view summarises what has transpired during the session – including the total number of analyses performed, the number of atrial fibrillation (AF) cases detected, and the number of normal rhythm cases – and provides two buttons that allow the physician to quickly access the ECG Analysis module or the patient archive.

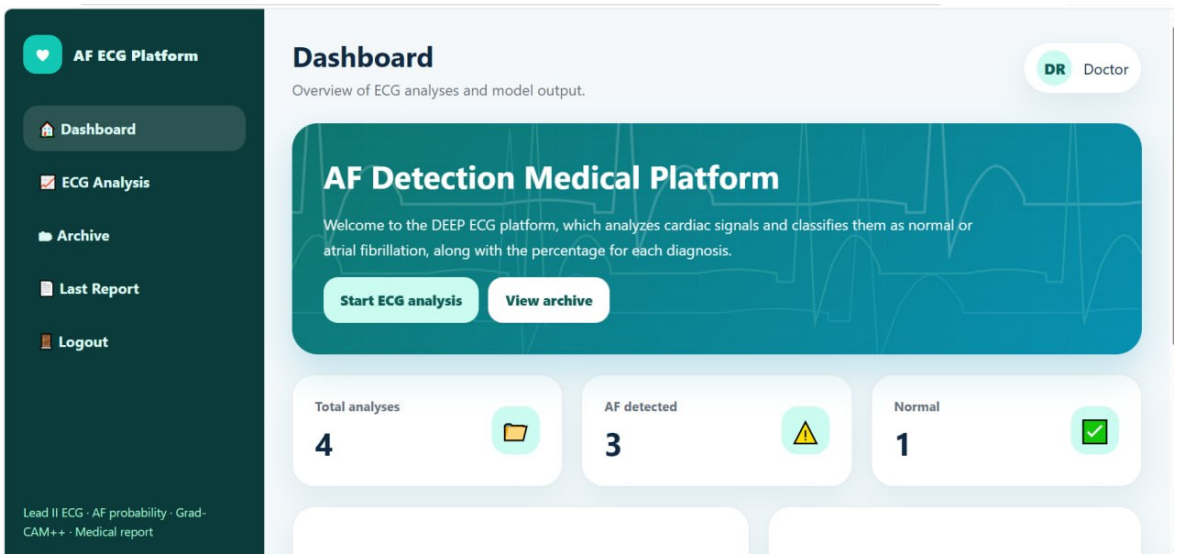


Figure 4.7: Dashboard overview displaying session statistics (total analyses, AF detected, Normal), pipeline description, and navigation controls.

ECG Analysis Module

The ECG Analysis Module is the primary module in the platform. To access this module, the doctor enters the patient name, uploads the Lead II ECG CSV file, and submits the file for analysis through the API.The interface then displays the AF probability and binary decision alongside the 10-beat ECG strip (coloured with P-wave markings), the AI explanation panel, and a clinical recommendation based on the results (as shown in fig. 4.8). The results of the analysis are also automatically stored in the archive.

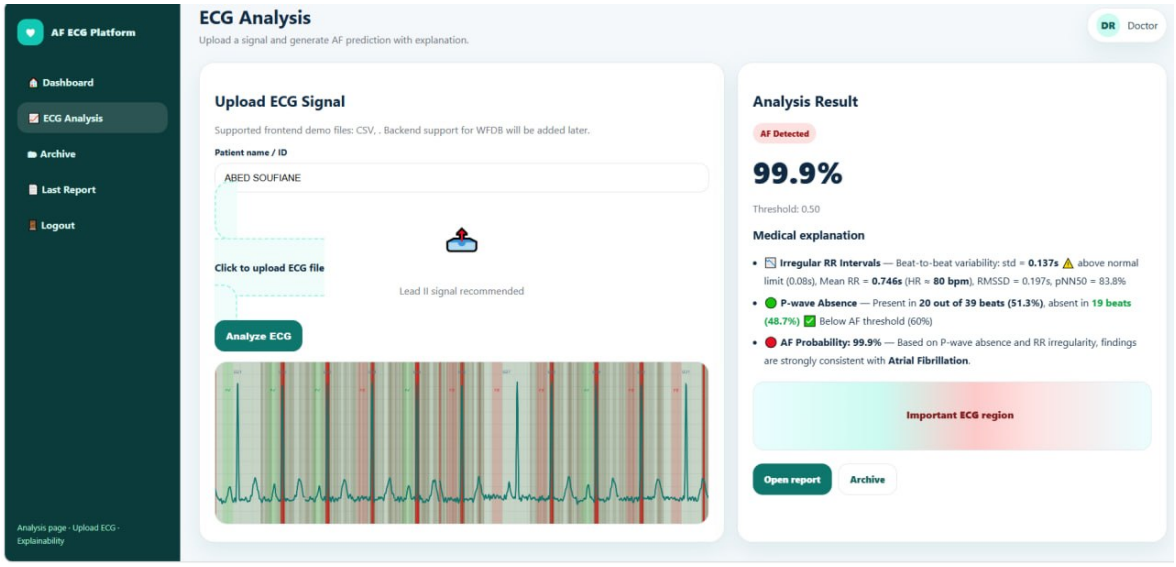


Figure 4.8: Interactive ECG Analysis interface displaying the input and signal upload panel, the processed Lead II ECG waveform with automated segment annotations, and the real-time AI classification result sidebar with medical explanation metrics.

Patient Archive

After several analyses have been performed, the doctor can review them collectively in the Archive module. As shown in Figure 4.9, each entry includes the patient name, ECG file name, timestamp, classification decision, and AF probability, and the doctor can review the complete report for any entry or delete it from the archive.

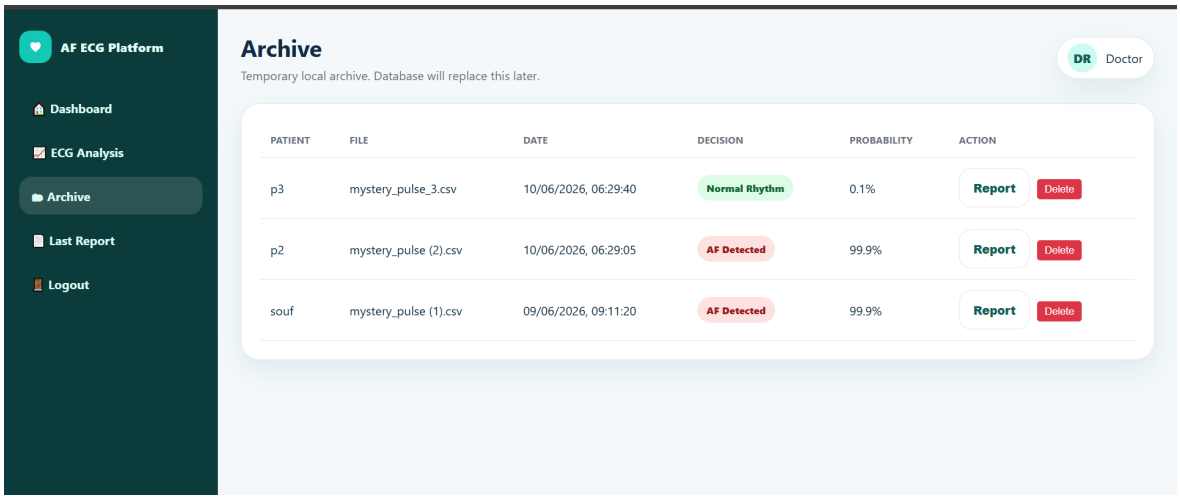


Figure 4.9: Patient Archive panel showing a chronological log of ECG analyses with AF probabilities, classification decisions, and report/delete actions.

Medical Report Generation

The Last Report section provides all output in a medical report format, which includes patient information, predicted diagnosis with probability, image of ECG, and detailed explanation panel. This report can be printed and/or saved as PDF directly through the browser. Each report has an AI disclaimer at the top, indicating that results are AI-assisted only, and should be validated by a clinician prior to making a clinical decision(as shown in fig. 4.10).

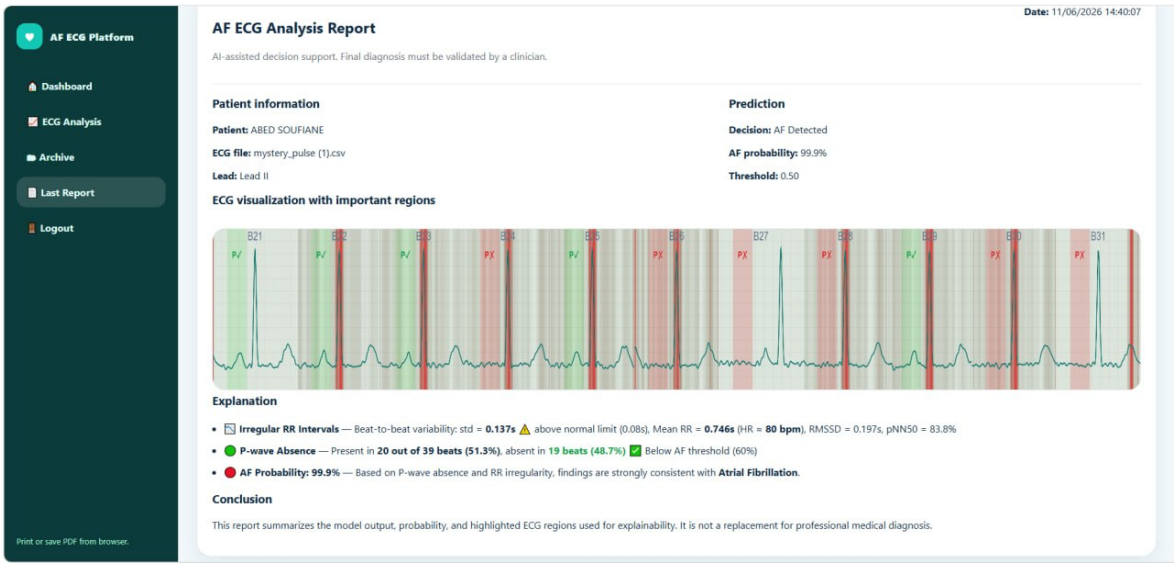


Figure 4.10: Comprehensive AI-generated clinical report interface within the Last Report module, illustrating patient diagnostics, automated parameter explanations, and browser-ready printing options.

4.5 Conclusion

This chapter presented a robust empirical evaluation and verification of the proposed Deep-ECG system compared to baseline options. The chapter began with a thorough description of the unified development environment, the multiple datasets, and the experimental arrangements to ensure that a valid and reproducible benchmark was established under similar training conditions.

The internal comparative evaluation validated the hypothesis that by jointly modelling localised heartbeat morphology and globally modelling RR-interval rhythm dynamics, richer feature representations are created. The two-branch architecture outperformed the one-branch baseline for each test, and the cross-attention fusion technique outperformed all other implemented fusion techniques, achieving peak diagnostic accuracy (99.18%) and F1-Score (98.59%) on the set-aside test set, with an AUC-ROC measure of 0.9950 and a very low false positive rate (0.39%). Additionally, when benchmarked against state-of-the-art literature that recently appeared, the Deep-ECG system demonstrates an exceptionally robust, highly sensitive, and highly selective diagnostic option for detecting single-lead Atrial Fibrillation. With exceptional classification accuracy and very low latencies, the proposed model is an ideal candidate for use in automated clinical screening and ambulatory monitoring in real-world clinical settings.Furthermore, the system's transparency was validated through explainability analysis, highlighting the exact morphological and rhythmic biomarkers driving the model's decisions. To bridge the gap between research and clinical utility, a functional web application was developed and deployed, demonstrating how these predictions and explanations can be visualised in real-time.

GENERAL CONCLUSION

The advancements in Atrial Fibrillation detection have witnessed significant progress with the integration of state-of-the-art deep learning techniques. The effectiveness of the proposed Deep-ECG system, combining a shared BeatCNN with two dedicated Transformer encoders and a cross-attention fusion module, was carefully investigated in this work. Our thorough analysis has shown how crucial this dual-stream approach is for precisely identifying the subtle patterns of Atrial Fibrillation — a capacity that is necessary for reliable clinical application.

We used a variety of different, well-annotated datasets for our research, including the CPSC2018, Georgia 12-Lead ECG Challenge, and ECG-Arrhythmia datasets from PhysioNet. These databases were essential for training and evaluating the model and for guaranteeing strong cross-scenario generalisation. According to the experimental results, the dual-branch architecture demonstrated outstanding performance, achieving an accuracy of 99.18%, an F1-score of 0.9859, and an AUC of 0.9950 on a test set of 1,823 recordings. These figures confirm that jointly exploiting beat morphology and RR rhythm dynamics yields a meaningful and consistent improvement over single-stream approaches.

Furthermore, the integration of masked mean pooling, positional encodings, and the cross-attention fusion mechanism significantly enhanced

the system's capacity to capture both local waveform shape and long-range temporal dependencies. These design choices highlight the model's flexibility and robustness in addressing the inherent difficulties of ECG classification, including signal noise, inter-patient variability, and class imbalance. The thorough evaluation of each architectural component offered insightful information about the design decisions that drive performance and will guide future improvements.

Future Work

To further extend the capabilities of the proposed system, the following directions are identified for future research:

- **Multi-class arrhythmia classification:** Extend the binary AF / Non-AF system to cover additional rhythm disorders such as atrial flutter, ventricular tachycardia, and bundle branch blocks, making Deep-ECG a comprehensive arrhythmia recognition tool.
- **Larger and more diverse datasets:** Curate and annotate a larger corpus covering different recording devices, patient demographics, and clinical environments to improve the model's generalisation and robustness across real-world scenarios.
- **Multi-lead input:** Incorporate multiple ECG leads through early fusion or a cross-lead attention layer to exploit the full spatial information available in standard 12-lead recordings.
- **Real-time wearable integration:** Adapt the beat segmentation and RR extraction pipeline to operate on continuous sliding-window streams from ambulatory wearable devices such as smartwatches and Holter patches, enabling real-time AF monitoring outside hospital settings.

- **Interpretability:** Visualise the attention weights of the Beat Transformer and the cross-attention fusion module to highlight which heartbeats and RR intervals drive each prediction, improving clinician trust in the system.
- **Lightweight model optimisation:** Apply quantisation, pruning, or knowledge distillation to reduce model size and inference time, enabling deployment on resource-constrained edge devices.

These improvements seek to bridge the gap between laboratory research and practical clinical deployment, reinforcing the efficacy and long-term impact of the Deep-ECG system.

BIBLIOGRAPHY

[1] Y. Xu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," *The Innovation*, vol. 2, no. 4, p. 100179, 2021. doi: 10.1016/j.xinn.2021.100179.

[2] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[3] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[4] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021. doi: 10.1007/s42979-021-00815-1.

[5] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, 2019. doi: 10.1007/s11831-019-09344-w.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[9] World Health Organization (WHO), "Cardiovascular diseases (CVDs)," [Online]. Available: https://www.who.int/en/health-topics/noncommunicable-diseases/cardiovascular-diseases. [Accessed: Apr. 1, 2026].

[10] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. San Diego, CA, USA: Academic Press, 2005.

[11] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000. doi: 10.1161/01.cir.101.23.e215.

[12] Y. Ansari *et al.*, "Deep learning for ECG arrhythmia detection and classification: An overview of progress for the period 2017–2023," *Frontiers in Physiology*, vol. 14, p. 1246746, Sep. 2023. doi: 10.3389/fphys.2023.1246746.

[13] Z. I. Attia *et al.*, "An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm," *The Lancet*, vol. 394, no. 10201, pp. 861–867, 2019. doi: 10.1016/S0140-6736(19)31721-0.

[14] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural

network," *Nature Medicine*, vol. 25, no. 1, pp. 65–69, Jan. 2019. doi: 10.1038/s41591-018-0268-3.

[15] T.-R. Wei, S. Lu, and Y. Yan, "Automated Atrial Fibrillation Detection with ECG," *Bioengineering*, vol. 9, p. 523, 2022. doi: 10.3390/bioengineering9100523.

[16] R. Hu, J. Chen, and L. Zhou, "A transformer-based deep neural network for arrhythmia detection using continuous ECG signals," *Computers in Biology and Medicine*, vol. 144, p. 105325, 2022. doi: 10.1016/j.compbiomed.2022.105325.

[17] Y. Ping *et al.*, "Automatic detection of atrial fibrillation based on CNN-LSTM and shortcut connection," *Healthcare*, vol. 8, no. 2, p. 139, 2020. doi: 10.3390/healthcare8020139.

[18] J. Xie, S. Stavrakis, and B. Yao, "Automated identification of atrial fibrillation from single-lead ECGs using multi-branching ResNet," *Frontiers in Physiology*, vol. 15, p. 1362185, 2024. doi: 10.3389/fphys.2024.1362185.

[19] P. Wang, J. Song, P. Wu, Z. Li, X. Chen, L. Du, and Z. Fang, "AF-DETR: Transformer-based object detection for precise atrial fibrillation beat localization in ECG," *Bioengineering*, vol. 12, no. 10, p. 1104, 2025. doi: 10.3390/bioengineering12101104.

[20] Python Software Foundation, *Python 3.12.13 documentation*, 2026. [Online]. Available: https://docs.python.org/3.12/

[21] PyTorch Foundation, *PyTorch documentation — PyTorch 2.10 documentation*. [Online]. Available: https://pytorch.org/docs/2.10/

[22] The SciPy community, *SciPy User Guide — SciPy v1.16.2 Manual*, 2008. [Online]. Available: https://docs.scipy.org/doc/scipy-1.16.2/

[23] D. Makowski *et al.*, *Overview — NeuroKit2 0.2.13.dev214 documentation*, 2026. [Online]. Available: https://neuropsychology.github.io/NeuroKit/

[24] MIT Lab for Computational Physiology, *wfdb — wfdb "4.3.1" documentation*, 2018. [Online]. Available: https://github.com/MIT-LCP/wfdb-python

[25] E. A. Perez Alday *et al.*, "Classification of 12-lead ECGs: The PhysioNet/Computing in Cardiology Challenge 2020," *PhysioNet*, version 1.0.2, 2022. doi: 10.13026/dvyd-kd57.

[26] J. Zheng, H. Guo, and H. Chu, "A large scale 12-lead electrocardiogram database for arrhythmia study," *PhysioNet*, version 1.0.0, 2022. doi: 10.13026/wgex-er52.